Available online at www.sciencedirect.com
ScienceDirect
International Journal of Approximate Reasoning
xxx (2008) xxx–xxx
INTERNATIONAL JOURNAL OF APPROXIMATE REASONING
www.elsevier.com/locate/ijar

Inference in directed evidential networks based on the transferable belief model

Boutheina Ben Yaghlane *, Khaled Mellouli

LARODEC, Université de Tunis, IHEC-Carthage Présidence 2016 Tunis, Tunisia

Received 9 July 2006; received in revised form 2 December 2007; accepted 3 January 2008

Abstract

Inference algorithms in directed evidential networks (DEVN) obtain their efficiency by making use of the represented independencies between variables in the model. This can be done using the *disjunctive rule of combination* (DRC) and the *generalized Bayesian theorem* (GBT), both proposed by Smets [Ph. Smets, Belief functions: the disjunctive rule of combination and the generalized Bayesian theorem, International Journal of Approximate Reasoning 9 (1993) 1–35]. These rules make possible the use of conditional belief functions for reasoning in directed evidential networks, avoiding the computations of joint belief function on the product space. In this paper, new algorithms based on these two rules are proposed for the propagation of belief functions in singly and multiply directed evidential networks.
© 2008 Elsevier Inc. All rights reserved.

Keywords: Belief functions; Conditional belief functions; Directed evidential networks; Binary join tree

1. Introduction

Graphical models are increasingly popular knowledge representations for reasoning under uncertainty. The success of these graphical models resides in their capacity for representing and handling conditional independence relationships, which facilitate efficient information management and storage. This notion of conditional independence is very important because it simplifies several computational reasoning tasks.

In the networks using belief functions, the relations among the variables are usually represented by joint belief functions rather than conditional belief functions. Recently, a number of techniques for handling conditional belief functions have been developed. Cano et al. [5] have presented an axiomatic system, based on Shenoy–Shafer's axiomatic for propagating uncertainty (including belief functions) in directed acyclic graphs. Smets [27] has proposed the generalized Bayesian theorem (GBT) and presented the disjunctive rule of combination (DRC) for two distinct pieces of evidence which make possible the representation of knowledge by conditional belief functions and their use for reasoning in belief networks. Shenoy [21] has explored the use of a graphical representation of valuation-based systems (VBS), called valuation networks, for representing

* Corresponding author. Tel.: +216 71 775 948; fax: +216 71 775 944.
E-mail addresses: boutheina.yaghlane@ihec.rnu.tn (B.B. Yaghlane), khaled.mellouli@ihec.rnu.tn (K. Mellouli).

0888-613X/$ - see front matter © 2008 Elsevier Inc. All rights reserved.
doi:10.1016/j.ijar.2008.01.002

Please cite this article in press as: B.B. Yaghlane, K. Mellouli, Inference in directed evidential networks based ..., Int. J. Approx. Reason. (2008), doi:10.1016/j.ijar.2008.01.002

UNCORRECTED PROOF

conditional relations. Xu and Smets [32] have presented an alternative framework to the general VBS, called evidential networks with conditional belief functions (ENC).

In order to motivate the use of conditional belief functions, we first introduce an example showing the relationships between joint belief functions and conditional belief functions which represent the same knowledge.

Example 1. Let X and Y be two variables formalizing the following two questions:

- Q_1: Does the patient have a disease D?
- Q_2: What is the result of the medical test T?

Let $\Theta_X = \{D, ND\}$ be the frame for X representing the fact that the patient has disease (D) or not (ND), and $\Theta_Y = \{+, -\}$ be the frame for Y representing the set of all possible answers to the medical test. It can be either positive or negative.

To represent a relation between X and Y such that: "If the patient has the disease D then the result of the medical test is positive with $m = 0.95$" by a belief function in:

Joint form: the rule is represented by a belief function on the space $\Theta = \Theta_X \times \Theta_Y = \{(D,+),(D,-),(ND,+),(ND,-)\}$, with masses: 0.95 on the subset $\{(D,+),(ND,+),(ND,-)\}$, and 0.05 on Θ.

Conditional form: the rule is represented by the conditional belief mass $m[X](Y)$:[1]

$$m[D](\{+\}) = 0.95,$$
$$m[D](\Theta_Y) = 0.05,$$
$$m[ND](\Theta_Y) = 1,$$
$$m[\Theta_X](\Theta_Y) = 1.$$

According to this example, it is possible to show that the conditional representation is more natural and easier for the users to provide and to understand than the joint representation. Moreover, given two variables X and Y on respective frames Θ_X and Θ_Y, to represent a conditional belief function for Y given X by a conditional form requires at most $2^{|\Theta_X|+|\Theta_Y|}$ values, while by a joint form, it requires $2^{|\Theta_X|\times|\Theta_Y|}$ values in the worst case, a serious gain once $|\Theta_X|$ and $|\Theta_Y|$ become large.

Thus, the aim of this paper is to provide new inference algorithms for improving the efficiency of belief function propagation in directed evidential networks with conditional belief functions. To improve the inference process, we avoid the computation of joint belief functions on the product space by using GBT and DRC rules proposed by Smets [27].

The remainder of this paper is organized as follows: In Section 2, we introduce the theory of belief functions and the transferable belief model (TBM). Then we discuss, in Section 3, some aspects related to uncertainty representation in directed evidential networks with conditional belief functions. Section 4 presents the necessary rules for reasoning directly with conditional belief functions and shows how to do efficient computation in DEVNs using the GBT and DRC. Section 5 presents a propagation algorithm extended from Pearl's polytree algorithm [16] and expressed in terms of belief functions. In Section 6, we deal with the general case of multi-directed evidential networks and propose an algorithm which directly manipulates the conditional belief functions, as opposed to first calculating joint belief functions, then we discuss some related computational aspects. Finally, in Section 7, we summarize the contributions achieved in this paper and we point out some future directions.

2. A review of the belief function theory

The theory of belief functions [18], also known as Dempster–Shafer theory or evidence theory, aims to model someone's degree of belief. It is regarded as a generalization of the Bayesian approach. Since this theory

[1] The conditional belief mass $m[X](Y)$ can be read as the mass of Y given X. Classically, it was denoted by $m(Y|X)$, but the bracket notation turns out to be more convenient.

Please cite this article in press as: B.B. Yaghlane, K. Mellouli, Inference in directed evidential networks based ..., Int. J. Approx. Reason. (2008), doi:10.1016/j.ijar.2008.01.002

UNCORRECTED PROOF

was developed, it has been followed by a number of theoretical and practical contributions giving different interpretations of the concept of *belief functions*. Among them, we can distinguish:

- the *lower probability model* where beliefs are represented by families of probability functions and the belief functions are the lower envelop of these families. This model is considered as a special case of imprecise probabilities [29].
- the *Dempster's model* [8] derived from probability theory where a probability space is mapped by a one-to-many mapping on another space. It is essentially at the core of the hint theory [13]. From practical view, the probabilistic argumentation systems model (PAS) [11] shows how belief functions are obtained from a simple way of combining classical propositional logic (or corresponding extensions) with probability theory.
- the *transferable belief model* (TBM) [28] where beliefs are represented by belief functions. This model is unrelated to probability models whereas the other two are a generalization of them.

In this paper, we adopt the terminology of the belief function as described in the transferable belief model (TBM) [28]. Classical material about belief functions and the TBM can be found in Shafer [18] and Smets [28].

The belief function theory was originally developed for a discrete set of elementary events related to a given problem [18]. This set is referred to the *frame of discernment* and is denoted by Θ. Beliefs are expressed on the subsets of Θ. The power set of Θ, denoted as 2^Θ, is a set containing all the possible subsets of Θ. The mapping $bel : 2^\Theta \to [0,1]$ is a *belief function* if and only if there exists a *basic belief assignment* (bba) $m : 2^\Theta \to [0,1]$ such that:

$$\sum_{A \subseteq \Theta} m(A) = 1,$$
$$bel(A) = \sum_{\emptyset \neq B \subseteq A} m(B), \quad \text{and} \quad bel(\emptyset) = 0,$$
$$pl(A) = \sum_{B \cap A \neq \emptyset} m(B), \quad \text{and} \quad pl(\emptyset) = 0.$$

The value $m(A)$ represents the degree of belief that is exactly committed to the subset A of Θ given a piece of evidence. Due to the lack of information, $m(A)$ cannot support any more specific event. When $m(\emptyset) = 1$, m is called a *vacuous* bba. When $m(\emptyset) = 0$, m is called a *normalized* bba. In the TBM, we accept that $m(\emptyset)$ can be positive, a condition that has been referred to as the open-world assumption [28]. The value $bel(A)$ quantifies the strength of the belief that the event A occurs. A subset A such as $m(A) > 0$ is called a *focal element* of *bel*. The value $pl(A)$ quantifies the maximum amount of potential specific support that could be given to A.

When we model aspects of the real world, we often deal with multivariate situations where the state space is a product space. We present below some definitions necessary when using belief functions.

2.1. Variables

Let $\mathbf{U} = \{X, Y, Z, \ldots\}$ be a set of finite variables, $\Theta_X = \{x_1, \ldots, x_n\}$ be the domain relative to the variable X (with a finite cardinality n), and x represents any instance of X. For $x \subseteq \Theta_X$ and $y \subseteq \Theta_Y$, (x,y) is defined by $(x,y) = \{(x_i, y_j) : x_i \in x, y_j \in y\}$.

For a subset of variables $A \subseteq \mathbf{U}$, the *frame for A*, denoted by Θ_A, is the Cartesian product of the frames for the variables in A, and its elements are called the *configurations* of A. Extension and projection of sets of configurations are very important in belief functions theory and are defined as follows:

Definition 1 (*Cylindrical extension*). For $x \subseteq \Theta_X$, $x^{\uparrow XY}$ is the *cylindrical extension of* x on $\Theta_X \times \Theta_Y$: $x^{\uparrow XY} = (x, \Theta_Y)$.

Definition 2 (*Projection*). For $w \subseteq \Theta$, $w^{\downarrow X}$ is the projection of w on X: $w^{\downarrow X} = \{x_i : x_i \in \Theta_X, x_i^{\uparrow \Theta} \cap w \neq \emptyset\}$.

UNCORRECTED PROOF

Please cite this article in press as: B.B. Yaghlane, K. Mellouli, Inference in directed evidential networks based ..., Int. J. Approx. Reason. (2008), doi:10.1016/j.ijar.2008.01.002

2.2. Belief function operations

2.2.1. Conditioning

Let x denote the background knowledge that holds and underlies the beliefs. In x, we find the classical conditioning events. We use the notation $m^{\Theta}[x]$ to represent the basic belief assignment defined on the domain Θ given the belief holder knows that x is true (i.e. x holds). The term m can be replaced by *bel* or *pl* in order to denote the *belief function* or the *plausibility function*.

Given a belief function bel^{Θ} on Θ, let

$$b^{\Theta}(w) = bel^{\Theta}(w) + m^{\Theta}(\emptyset) \quad \forall w \subseteq \Theta.$$

This function b^{Θ} is called the *implicability function*. In practice, the b function is much more convenient to work with than the *bel* function.

The value of *conditional belief function* at $w \subseteq \Theta$ is denoted by $bel^{\Theta}[x](w)$. It represents the belief of w given x. When x is the proposition stating that the value of Θ belongs to $y \subseteq \Theta$, its value is given by

$$b^{\Theta}[y](w) = b^{\Theta}(w \cup \bar{y}),$$
$$bel^{\Theta}[y](w) = bel^{\Theta}(w \cup \bar{y}) - bel^{\Theta}(\bar{y}),$$
$$pl^{\Theta}[y](w) = pl^{\Theta}(w \cap y).$$

This is the so-called *Dempster's rule of conditioning* (except for the normalization factor).

2.2.2. Combination

The $\oplus$ symbol represents *Dempster's rule of combination* in its normalized form and $\textcircled{\cap}$ represents the *conjunctive combination*, i.e., the same operation as Dempster's rule of combination except the normalization is not performed.

The conjunctive combination rule (as well as its Dempster' form) are applicable to combine the belief functions produced by distinct pieces of evidence and can be written as

$$(m_1 \textcircled{\cap} m_2)(w) = \sum_{w_1, w_2 \subseteq \Theta, w_1 \cap w_2 = w} m_1(w_1) m_2(w_2).$$

The next formula is very useful

$$f_{1\textcircled{\cap}2}(w) = \sum_{w^* \subseteq \Theta} f_1[w^*](w) m_2(w^*) \quad \forall w \subseteq \Theta, \tag{1}$$

where $f \in \{m, bel, pl\}$ and $f_1[w^*]$ is the result of the unnormalized conditioning of f_1 on $w^* \subseteq \Theta$ (see [27]).

The conjunctive combination rule can equivalently and very conveniently be written as follows: for all $A \subseteq \Theta$,

$$m_{1\textcircled{\cap}2}(A) = \sum_{B \subseteq \Theta} m_1[B](A)\ m_2(B), \tag{2}$$

$$b_{1\textcircled{\cap}2}(A) = \sum_{B \subseteq \Theta} b_1[B](A)\ m_2(B), \tag{3}$$

$$pl_{1\textcircled{\cap}2}(A) = \sum_{B \subseteq \Theta} pl_1[B](A)\ m_2(B). \tag{4}$$

2.2.3. Marginalization

The belief function $bel^{Y\downarrow X}$ (where $X \subseteq Y \subseteq \mathbf{U}$, $X \neq \emptyset$) is the marginal of bel^{Y} on X. In particular, we have:

$$bel^{Y\downarrow X}(x) = bel^{XY}(x, \Theta_Y),$$
$$pl^{Y\downarrow X}(x) = pl^{XY}(x, \Theta_Y).$$

Conditioning and marginalization do not commute, so the order of the symbols is important. The belief function obtained by conditioning bel^{XY} on y and then marginalizing the result on X is given by $bel^{XY}[y]^{\downarrow X}$.

UNCORRECTED PROOF

Please cite this article in press as: B.B. Yaghlane, K. Mellouli, Inference in directed evidential networks based ..., Int. J. Approx. Reason. (2008), doi:10.1016/j.ijar.2008.01.002

2.2.4. Ballooning extension

Let X and Y be two disjoint subsets of $\mathbf{U}$, and let $bel^X[y_i]$ be a conditional belief function defined on Θ_X for $y_i \in \Theta_Y$. The ballooning extension of the conditional belief function, denoted $m^{X\uparrow XY}$, is the belief function defined on $\Theta_X \times \Theta_Y$ whose bba satisfies [26]:

$$m^{X\uparrow XY}(w) = \begin{cases} m^X[y_i](x) & \text{if } w = (x, y_i) \cup (X, \overline{y_i}), \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

3. Directed evidential networks with conditional belief functions

In this section, we discuss some aspects related to uncertainty representation in directed evidential networks with conditional belief functions. Pearl starts with conditional independence relationships when building his *probabilistic graphical model* where conditional probabilities can be directly manipulated using Bayes' theorem [16]. This graphical model is considered as a directed acyclic graph that provides an intuitive qualitative and quantitative description of the problem. It is also considered as a mathematical structure that specifies the different connections between the variables of a complex problem transforming it into a clear representation.

However, in the *networks using belief functions*, the relations among the variables are usually represented by joint belief functions rather than conditional belief functions. Furthermore, these networks are undirected graphs and presented, in different contexts, by *join trees* [25], *qualitative Markov trees* [19], *hypertrees* [24], or more generally in the framework of *valuation-based system* and its graphical representation *valuation networks* [20].

Nevertheless, the use of graphs to represent conditional belief function independence relations [2,3] is useful since an exponential number of conditional independence statements can be represented by a graph with a polynomial number of vertices [21]. Due to the lack of directed belief networks (similar to Bayesian networks), recent works integrate belief function theory and Bayesian networks. We cite, for example, the work of Cobb and Shenoy [6] in which the authors propose a plausibility transformation method for translating Dempster–Shafer belief function models to probability models, and the work of Simon and Weber [17] in which the authors present an implementation of the Dempster–Shafer theory in a Bayesian Network tool in order to model reliability uncertainty.

In this paper, we show that it is possible to represent knowledge directly in the form of conditional belief functions. For this purpose, we focus on directed networks using belief functions and we introduce the *directed evidential networks with conditional belief functions*. Full description of this structure and its link with Bayesian networks and valuation networks can be found in [4].

3.1. Calculating joint belief functions

Let X and Y be two disjoint subsets of $\mathbf{U}$, and let the set of conditional belief functions $bel^X[y_i]$ be defined on Θ_X for each $y_i \in \Theta_Y$. The conditional belief functions are considered as produced by distinct pieces of evidence. In practice, this means that the knowledge of the value of $bel^X[y_i]$ does not produce constraints on the values of $bel^X[y_j]$ for $j \neq i$.

We want to build a bba on $\Theta_X \times \Theta_Y$ such that its conditioning on (X, y_i) reproduces $bel^X[y_i]$ when the conditional belief functions are generated by distinct pieces of evidence. The solution, presented in [26,27], is obtained as follows: "Each $bel^X[y_i]$ is extended by a ballooning extension on the frame $\Theta_X \times \Theta_Y$ by relation (5) and the results are conjunctively combined". The value of the resulting belief function is given in the following theorem.

Theorem 1. *Let X and Y be two disjoint subsets of $\mathbf{U}$, and let the set of conditional belief functions $bel^X[y_i]$ be defined on Θ_X for each $y_i \in \Theta_Y$. Let $w \subseteq \Theta_X \times \Theta_Y$. For each $y_i \in \Theta_Y$, let $x_i^X(w)$ be the projection on X of the elements of w which intersect (X, y_i): $x_i^X(w) = (w \cap (X, y_i))^{\downarrow X}$. Then the conjunctive combination of the ballooning extensions of each conditional belief function on $\Theta_X \times \Theta_Y$, $m^{XY} = \textcircled{\cap}_{y_i \in \Theta_Y} m^X[y_i]^{\uparrow XY}$, admits the following representations*:

Please cite this article in press as: B.B. Yaghlane, K. Mellouli, Inference in directed evidential networks based ..., Int. J. Approx. Reason. (2008), doi:10.1016/j.ijar.2008.01.002

UNCORRECTED PROOF

$$m^{XY}(w) = \prod_{y_i \in \Theta_Y} m^X[y_i](x_i^X(w)), \tag{6}$$

$$b^{XY}(w) = \prod_{y_i \in \Theta_Y} b^X[y_i](x_i^X(w)), \tag{7}$$

$$pl^{XY}(w) = 1 - \prod_{y_i \in \Theta_Y} (1 - pl^X[y_i](x_i^X(w))). \tag{8}$$

The formulas of Theorem 1 are proved in [26,27].

This construction is linked to the concept of a joint belief function with a vacuous marginal by the next lemma. The joint belief function built according to Theorem 1 is a joint belief function with a vacuous marginal over Y once all the conditional belief functions are normalized.

Lemma 1. *Let* $\{bel^X[y] : y \in \Theta_Y\}$ *be a family of normalized conditional belief functions for* X *given* Y. *Then* $bel^{XY} = \textcircled{\scriptsize}_{y \in \Theta_Y} bel^X[y]^{\uparrow XY}$ *is a joint belief function with a vacuous marginal over* Y.

3.2. Evidential networks with conditional belief functions

Evidential networks with conditional belief functions (ENC) were originally proposed by Smets for the propagation of beliefs [27], and later studied by Xu [31]. ENCs contain a directed acyclic graph with conditional beliefs defined in a different manner from conditional probabilities in Bayesian networks (BNs): each edge represents a conditional relation between the two nodes it connects. For example, the edges (X,Z) and (Y,Z) in (Fig. 1) mean that we have $\{bel^Z[x_i] : x_i \in \Theta_X\}$ and $\{bel^Z[y_i] : y_i \in \Theta_Y\}$, but not $\{bel^Z[x_i,y_i] : x_i \in \Theta_X, y_i \in \Theta_Y\}$ as in a BN.

However, if conditional beliefs such as $\{bel^Z[x_i,y_i] : x_i \in \Theta_X, y_i \in \Theta_Y\}$ are given, Xu's method builds an ENC in which nodes X and Y are merged as one node. For any merged node, the belief function is obtained by combining the ballooning extension of each conditional belief function.

3.3. Directed evidential network model

In the previous section, we have presented Xu's graphical representation (i.e. ENC) which uses conditional belief functions for knowledge representation and reasoning. By comparing some relations between the representation by joint beliefs and by conditional beliefs, Xu and Smets [32] have shown that the conditional form takes less space. Indeed, in ENCs, any computations involving two connected variables X and Y are processed on the space Θ_X or Θ_Y, while in the network with joint beliefs, such computations are always done on the product space $\Theta_X \times \Theta_Y$. Thus the computations in an ENC needs fewer set-comparisons and multiplications than networks with joint belief functions.

Despite some advantages, computational efficiency in ENCs can still be cumbersome due to reasoning process based on the ballooning extension. Furthermore, the representation and propagation algorithm proposed by Xu are restricted because they are for evidential networks which only have binary relations between the nodes [31].

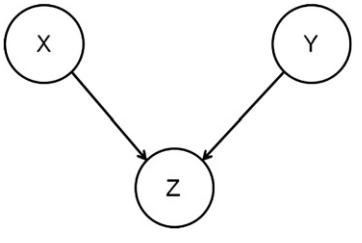


Fig. 1. An evidential network with conditional belief functions.

UNCORRECTED PROOF

Please cite this article in press as: B.B. Yaghlane, K. Mellouli, Inference in directed evidential networks based ..., Int. J. Approx. Reason. (2008), doi:10.1016/j.ijar.2008.01.002

Thus, in this section, we generalize ENCs to the case where relations are for any number of nodes. In order to distinguish the two kinds of networks, we call ours DEVN which means a *directed evidential network with conditional belief functions* and represented by

1. *A knowledge base*: we can distinguish two levels: qualitative and quantitative. At *the qualitative level*, we have a *directed acyclic graph* (DAG) in which the nodes represent variables, and directed arcs describe the conditional dependence relations embedded in the model. These dependence relations are expressed by *conditional belief functions* for each node given its parents.
2. *Facts*: represent the new observations introduced in the network and are represented by belief functions allocated to some nodes.

Each node X in the DEVN has a set of possible values called *frame of discernment* which consists of mutually exclusive and exhaustive values. Parents of X are denoted by Pa_X. For each root node X ($Pa_X = \emptyset$), uncertainty is represented by an *a priori belief function* bel_0^X on X. For the other nodes (i.e. $Pa_X \neq \emptyset$), uncertainty is represented by a *conditional belief function* $bel^X[Pa_X]$ on X given the value taken by its parents.

4. Efficient computation using GBT and DRC

Given an evidential system defined by a set of variables $\mathbf{U} = \{X_1, \ldots, X_n\}$ where each X_i of $\mathbf{U}$ takes its values on a frame Θ_{X_i}. Let $bel^{X_1}, \ldots, bel^{X_n}$ be the corresponding belief functions. We assume that for each $I \subseteq \{1, \ldots, n\}$ there is a set of belief functions bel^I defined on the cartesian product Θ_I. Then, we make inference by computing, for each variable X_i, the *marginal of joint belief function* $(bel^{X_1} \oplus \cdots \oplus bel^{X_n})^{\downarrow i}$. However, when there are many variables to manipulate, the computation of the joint belief functions becomes not feasible since the joint frame of variables is too large.

Many researchers have studied this problem so as to propose techniques for computing marginals without explicitly calculating the global belief function. A well-known method is the *local computation* among the initial belief functions in Markov trees [19]. The idea proposed by Shenoy and Shafer [25] is that if combination and marginalization operations verify three axioms, then the local computation becomes possible. Three new axioms are added by Cano et al. [5] to Shenoy–Shafer's axiomatic framework for the propagation in directed graphs. These axioms allow the definition of some aspects related with directed graphs.

In the following section, we provide the necessary tools to use belief functions in directed graphs (Sections 4.1 and 4.2). Then, we explain in Section 4.3 how to propagate any belief function in the simplest network with only three sets of variables. Finally, in Section 4.4 we present the case of the propagation with conditional beliefs.

4.1. The generalized bayesian theorem (GBT)

The *generalized Bayesian theorem* (GBT), proposed by Smets [27], is a generalization of Bayes' theorem where all conditional probabilities are replaced by belief functions. The main point is that the needed prior can be a vacuous belief function representing total ignorance.

Formally, let X and Y be two disjoint subsets of $\mathbf{U}$ defined on the spaces Θ_X and Θ_Y, respectively. We suppose that the additional information is given by $bel^Y[x_i]$ representing a conditional belief function induced on the space Θ_Y given x_i element of Θ_X. We want to compute $bel^X[y](x)$ for any $x \subseteq \Theta_X$ and $y \subseteq \Theta_Y$. This belief is derived from *the generalized Bayesian theorem* (GBT).

For given belief function $bel : 2^\Theta \to [0,1]$, Smets [27] defines a function $b : 2^\Theta \to [0,1]$ such that: $b(A) = bel(A) + m(\emptyset)$.

For any $x \subseteq \Theta_X$ and $y \subseteq \Theta_Y$, the GBT allows us to build the belief function $bel^X[y](x)$ as follows:

$$bel^X[y](x) = b^X[y](x) - b^X[y](\emptyset), \tag{9}$$

$$b^X[y](x) = \prod_{x_i \in \bar{x}} b^Y[x_i](\bar{y}). \tag{10}$$

INCORRECTED PROOF

Please cite this article in press as: B.B. Yaghlane, K. Mellouli, Inference in directed evidential networks based ..., Int. J. Approx. Reason. (2008), doi:10.1016/j.ijar.2008.01.002

If each $bel^Y[x_i]$ happens to be a probability function $P[x_i]$ on Θ_Y and the prior belief on Θ_X is also a probability function $P_0(x)$, then the normalized GBT is reduced into the classic Bayes' theorem (which explains the origin of its name):

$$m[y](x_i) = \frac{P_0(x_i) \times P[x_i](y)}{\sum_{x_i \in \Theta_X} P_0(x_i) \times P[x_i](y)} = P[y](x_i). \tag{11}$$

4.2. The disjunctive rule of combination (DRC)

Simultaneously, the belief function induced by the set of conditional belief functions on XY can be conditioned on $y \subseteq \Theta_Y$ and the result marginalized on X, corresponding to the so-called *disjunctive rule of combination* (DRC), also proposed by Smets [27].

Thus, if we want to compute $bel^Y[x](y)$ for any $x \subseteq \Theta_X$ and $y \subseteq \Theta_Y$, we use the DRC which allows us to build the belief function $bel^Y[x](y)$ as follows:

$$bel^Y[x](y) = b^Y[x](y) - b^Y[x](\emptyset), \tag{12}$$

$$b^Y[x](y) = \prod_{x_i \in x} b^Y[x_i](y). \tag{13}$$

When the belief functions are probability functions, they give $P^Y[x],\ x \subseteq \Theta_X$, from the $P^Y[x_i]$'s.

4.3. Simple propagation with any belief function

Let A, B, C be three mutually exclusive sets of variables. We assume that we have three nodes with the domains being $\Theta_{A\cup B}$, $\Theta_{B\cup C}$, $\Theta_{A\cup B\cup C}$ and two bba's m_1^{AB} and m_2^{ABC}. We want to transfer m_1 through m_2 and compute the resulting m_3^{BC} (see Fig. 2).

We must thus compute

$$m_3^{BC} = m_{12}^{ABC\downarrow BC}, \tag{14}$$

where

$$m_{12}^{ABC} = m_1^{AB\uparrow ABC} \textcircled{\cap} m_2^{ABC}.$$

The computation is simplified when we express the relation (14) with the implicability function. We have:

$$b_3^{BC}(v) = b_{12}^{ABC}(A, v) \quad \forall v \subseteq \Theta_{B\cup C}, \tag{15}$$

we get from relation (3) for all $v \subseteq \Theta_{B\cup C}$,

$$b_3^{BC}(v) = \sum_{u \subseteq \Theta_{A\cup B}} b_2^{ABC}[(u, C)](A, v) m_1^{AB}(u). \tag{16}$$

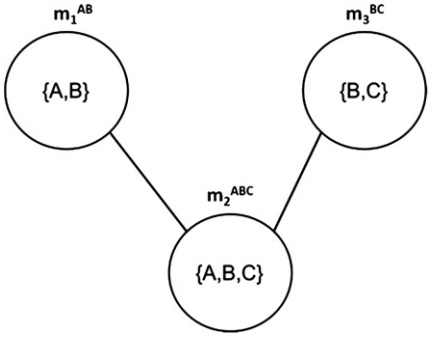


Fig. 2. Simple propagation.

Please cite this article in press as: B.B. Yaghlane, K. Mellouli, Inference in directed evidential networks based ..., Int. J. Approx. Reason. (2008), doi:10.1016/j.ijar.2008.01.002

CORRECTED PROOF

4.4. Propagation with conditional beliefs

Consider the directed evidential network with two nodes X and Y defined on frames Θ_X and Θ_Y, respectively. Suppose that there exists some a priori information over Θ_X given by belief function bel_0^X (m_0^X is its corresponding mass function) and some a priori information over Θ_Y given by $bel_0^Y(m_0^Y)$. We assume that we also have conditional belief functions $\{bel^Y[x_i] : x_i \in \Theta_X\}$.

For each node in the network, the marginal is computed by combining all the messages received from its neighbors and its own prior belief. So, if we want to compute bel^X of the node X, which is the parent of Y, we combine its prior belief bel_0^X with the message coming from Y, i.e.

$$bel^X = bel_0^X \oplus bel_{Y\to X}, \tag{17}$$

where $bel_{Y\to X}$ is a belief function on X representing the message coming from Y, and is computed by

$$\forall x \subseteq \Theta_X, \quad bel_{Y\to X}(x) = \sum_{y\subseteq\Theta_Y} m_0^Y(y) bel^X[y](x), \tag{18}$$

so that $bel^X[y](x)$ is the *posterior belief function* given by the generalized Bayesian theorem (using Eqs. (9) and (10)).

On the other hand, if we want to compute bel^Y of the node Y, which is the child of X, we combine its prior belief bel_0^Y with the message coming from X, i.e.

$$bel^Y = bel_0^Y \oplus bel_{X\to Y}, \tag{19}$$

where $bel_{X\to Y}$ is a belief function on Y representing the message coming from X, and is computed by

$$\forall y \subseteq \Theta_Y, \quad bel_{X\to Y}(y) = \sum_{x\subseteq\Theta_X} m_0^X(x) bel^Y[x](y), \tag{20}$$

so that $bel^Y[x](y)$ is the *conditional belief function* representing the relation between X and Y, and given by the disjunctive rule of combination (using Eqs. (12) and (13)).

Example 2. Consider the same variables X and Y and the same questions Q_1 and Q_2 as in Example 1. In order to represent our prior evidence about the answer to questions Q_1 and Q_2, let bel_0^X and bel_0^Y be two prior belief functions over Θ_X and Θ_Y, respectively, and be given by their respective mass functions m_0^X and m_0^Y as follows:

$$m_0^X(\{D\}) = 0.6, \quad m_0^Y(\{+\}) = 0.7,$$
$$m_0^X(\{ND\}) = 0.3, \quad m_0^Y(\{-\}) = 0.1,$$
$$m_0^X(\Theta_X) = 0.1, \quad m_0^Y(\Theta_Y) = 0.2.$$

To represent a relation between Θ_X and Θ_Y such as: "If the patient has the disease D then the result of the medical test is positive" with a belief of 0.95, and "If the patient does not have the disease D then the result of the medical test is negative" with a belief of 0.92, we use belief functions in conditional form. Then, the rules are represented by the following conditional mass functions m:

$$m[\{D\}](\{+\}) = 0.95, \quad m[\{ND\}](\{-\}) = 0.92.$$
$$m[\{D\}](\Theta_Y) = 0.05, \quad m[\{ND\}](\Theta_Y) = 0.08.$$

- Suppose that we want to find the resulting belief for question Q_1.

  Step 1: We first compute $bel_{Y\to X}$ which is a belief function computed by combining all conditional belief functions with its own a priori bel_0^Y (except X, the node Y has no other neighbors) using zFormula (18).

  For any $x \subseteq \Theta_X$, the belief function $bel_{Y\to X}(x)$ is given by its mass function $m_{Y\to X}(x)$ as follows:

$$m_{Y\to X}(\{D\}) = 0.7 \times 0.92 = 0.644,$$
$$m_{Y\to X}(\{ND\}) = 0.1 \times 0.95 = 0.095,$$
$$m_{Y\to X}(\Theta_X) = 0.7 \times 0.08 + 0.1 \times 0.05 + 0.2 \times 1 = 0.261.$$

UNCORRECTED PROOF

Please cite this article in press as: B.B. Yaghlane, K. Mellouli, Inference in directed evidential networks based ..., Int. J. Approx. Reason. (2008), doi:10.1016/j.ijar.2008.01.002

Step 2: Now, node X computes $bel_0^X \oplus bel_{Y\to X}$ using formula (17). The results are obtained by the application of the Dempster's rule of combination applied to the a priori belief (bel_0^X) and the message from Y ($bel_{Y\to X}$). The presented belief functions are normalized.

$$m_X(\{D\}) = 0.6074/0.7498 = 0.81,$$
$$m_X(\{ND\}) = 0.1163/0.7498 = 0.1552,$$
$$m_X(\Theta_X) = 0.0261/0.7498 = 0.0348,$$

where $k = 1 - (0.057 + 0.1932) = 0.7498$ is a normalization factor.

- Suppose that we want to find the resulting belief for question Q_2.

Step 1: We first compute $bel_{X\to Y}$ which is a belief function computed by combining all conditional belief functions with its own a priori bel_0^X using formula (20).

For any $y \subseteq \Theta_Y$, the belief function $bel_{X\to Y}(y)$ is given by its mass function $m_{X\to Y}(y)$ as follows:

$$m_{X\to Y}(\{+\}) = 0.6 \times 0.95 = 0.57,$$
$$m_{X\to Y}(\{-\}) = 0.3 \times 0.92 = 0.276,$$
$$m_{X\to Y}(\Theta_Y) = 0.6 \times 0.05 + 0.3 \times 0.08 + 0.1 \times 1 = 0.154.$$

Step 2: Now, node Y computes $bel_0^Y \oplus bel_{X\to Y}$ using formula (19) (Table 1)The results are obtained by the application of the Dempster's rule of combination applied to the a priori belief (bel_0^Y) and the message from X ($bel_{X\to Y}$). The presented belief functions are normalized.

$$m_Y(\{+\}) = 0.6208/0.7498 = 0.828,$$
$$m_Y(\{-\}) = 0.0982/0.7498 = 0.131,$$
$$m_Y(\Theta_Y) = 0.0308/0.7498 = 0.041,$$

where $k = 0.7498$ is a normalization factor (Table 2).

This example shows how, in evidential networks with conditional belief functions, the computations involving two connected variables X and Y are processed on the frame Θ_X or Θ_Y, while in the network with joint

Table 1
Combination of bel_0^X and $bel_{Y\to X}$

	$\{D\}$	$\{ND\}$	Θ_X
	0.644	0.095	0.261
$\{D\}$ 0.6	$\{D\}$ 0.3864	$\emptyset$ 0.057	$\{D\}$ 0.1566
{ND} 0.3	$\emptyset$ 0.1932	{ND} 0.0285	{ND} 0.0783
Θ_X 0.1	$\{D\}$ 0.0644	$\{ND\}$ 0.0095	Θ_X 0.0261

Table 2
Combination of bel_0^Y and $bel_{X\to Y}$

	$\{+\}$	$\{-\}$	Θ_Y
	0.57	0.276	0.154
$\{+\}$ 0.7	$\{+\}$ 0.399	$\emptyset$ 0.1932	$\{+\}$ 0.1078
$\{-\}$ 0.1	$\emptyset$ 0.057	$\{-\}$ 0.0276	$\{-\}$ 0.0154
Θ_Y 0.2	$\{+\}$ 0.144	$\{-\}$ 0.0552	Θ_Y 0.0308

Please cite this article in press as: B.B. Yaghlane, K. Mellouli, Inference in directed evidential networks based ..., Int. J. Approx. Reason. (2008), doi:10.1016/j.ijar.2008.01.002

UNCORRECTED PROOF

beliefs, such computations are always on the product space $\Theta_X \times \Theta_Y$. The direct consequence from this result is that the computations are faster and require less memory.

5. Propagation algorithm for singly directed evidential networks

When the graphical models representing the independence relationships corresponding to a given domain of knowledge are very dense, the processes needed to use them for inference tasks (propagation) may be time-consuming [14,16]. Some simplified models, such as *singly directed networks*[2] (also called *polytrees*), may relieve these problems at the expense of losing some representation capabilities, because the kind of independence relationships that may be represented are more restricted for singly connected networks than for general multi-connected networks [7]. Nevertheless, using singly directed networks, we gain in efficiency and simplicity in the procedures for propagating information.

For singly connected networks, the fundamental algorithm for exact probabilistic inference was proposed by Kim and Pearl [12]. The main principle underlying Pearl's polytree algorithm is the use of the probabilistic independence properties implicit in the structure of these networks. By exploiting these independence properties, the joint probability computation (over the global state space) can be factored into a number of *local computations* (over smaller state spaces) which communicate by message-passing between neighboring variables.

In this section, we will propose an efficient propagation algorithm for directed evidential networks with conditional belief functions. It consists of extending Pearl's algorithm [16] for belief functions. We will especially show how to do efficient computations using the GBT and the DRC rules.

For the need of the propagation algorithm, let $\mathbf{U}$ be a finite set of nodes of a given network. For each $X \in \mathbf{U}$, let $Pa(X) \subseteq \mathbf{U}$ be the set of parents of X, and $Ch(X) \subseteq \mathbf{U}$ be the set of children of X. For each node X, we store the available a priori belief function bel_0^X. We store also the conditional belief functions for each node given its parents $\{bel^X[Y] : Y \in Pa(X)\}$. Whenever a node X is instantiated, we have a new observation (O_X), otherwise O_X is a vacuous belief function. Like in Pearl's notations [16], each variable of the network has a λ *value* and π *value* associated with it. In addition, each variable passes a λ *message* to each of its parents and a π *message* to each of its children.

5.1. Initialization process

In this step, we show how to compute the *a priori* belief functions of each variable (node) of the network using the propagation method.

Algorithm. *Bel-Initialization.*

1. For each node X:
   - set $bel^X \leftarrow bel_0^X$
   - set $\pi_X \leftarrow bel^X$
   - set $\lambda_X \leftarrow$ *vacuous belief* ($*$ Representing total ignorance $*$).

2. For each root node X:
   - send a new $\pi_{X \to Y}$ message for all children Y of X using Eq. (20).

3. Node Y has to wait until it holds the messages of all its parents, then
   - compute the new π_Y value using this formula

     $$\pi_Y = bel^Y \oplus \left(\bigoplus_{X \in Pa(Y)} \pi_{X \to Y}\right),$$

   - compute the new marginal belief $bel^Y \leftarrow \pi_Y \oplus \lambda_Y$,
   - send a new π_Y message for all its children using Eq. (20).

[2] They are graphs where no more than one (undirected) path connects every two nodes.

Please cite this article in press as: B.B. Yaghlane, K. Mellouli, Inference in directed evidential networks based ..., Int. J. Approx. Reason. (2008), doi:10.1016/j.ijar.2008.01.002

UNCORRECTED PROOF

5.2. Updating process

When a new observation is introduced at a given node, the updating algorithm will be performed. This propagation algorithm can be regarded as a message-passing scheme. Indeed, its general idea is based on the fact that when we arrive at a node X, we compute all the *incoming messages*, and then we calculate π_X value, λ_X value, new marginal bel^X, and all *outgoing messages*. Notice that one node at a time is updated.

5.2.1. Computing values

For each node X of $\mathbf{U}$, we compute the marginal bel^X value representing the marginal belief of node X. It is obtained by combining its initial value with the new observation (O_X) and with the messages coming from all its parents and all its children:

$$bel^X = \pi_X \oplus \lambda_X, \tag{21}$$

where

$$\pi_X = bel_0^X \oplus \left(\underset{Y \in Pa(X)}{\oplus} \pi_{Y \to X} \right) \quad \text{and} \quad \lambda_X = O_X \oplus \left(\underset{Z \in Ch(X)}{\oplus} \lambda_{Z \to X} \right). \tag{22}$$

5.2.2. Computing messages

Once the node X is updated, it will send new messages to all its neighbors that have not yet been updated. These messages are computed as follows:

- $\pi_{X \to Y}$ representing the message sent from a node X to its children Y (Eq. (20)):

$$\pi_{X \to Y} = bel_{X \to Y} \quad \text{where } bel_{X \to Y}(y) = \sum_{x \subseteq \Theta_X} m^X(x) bel^Y[x](y), \tag{23}$$

so that $bel^Y[x](y)$ is given by the DRC (Eqs. (12) and (13)).

- $\lambda_{Y \to X}$ representing the message sent from a node Y to its parents X (Eq. (18)):

$$\lambda_{Y \to X} = bel_{Y \to X} \quad \text{where } bel_{Y \to X}(x) = \sum_{y \subseteq \Theta_Y} m^Y(y) bel^X[y](x), \tag{24}$$

so that $bel^X[y](x)$ is a posterior belief function given by the GBT (Eqs. (9) and (10)).

5.2.3. Propagation algorithm

When a new observation (O_X) is introduced at node X, we propose the following recursive algorithm to perform the updating:

Algorithm. *Bel-Updating*.

1. X computes its new value bel^X, using Eq. (21).
2. For every child node $Z \in Ch(X)$, we calculate and send the new message $\pi_{X \to Y}$ using Eq. (23) to all children not yet been updated.
3. For every parent node $Y \in Pa(X)$, we compute and send the new message $\lambda_{X \to Y}$ using Eq. (24) to all parents not yet been updated.
4. Then, we select a new node X of $\mathbf{U}$.

This propagation algorithm ends when there are no nodes to update.

Example 3. Suppose that we have the following simple belief network constituted by the nodes A, B, and C. After performing initialization, the state of the network is given by Fig. 3.

Please cite this article in press as: B.B. Yaghlane, K. Mellouli, Inference in directed evidential networks based ..., Int. J. Approx. Reason. (2008), doi:10.1016/j.ijar.2008.01.002

UNCORRECTED PROOF

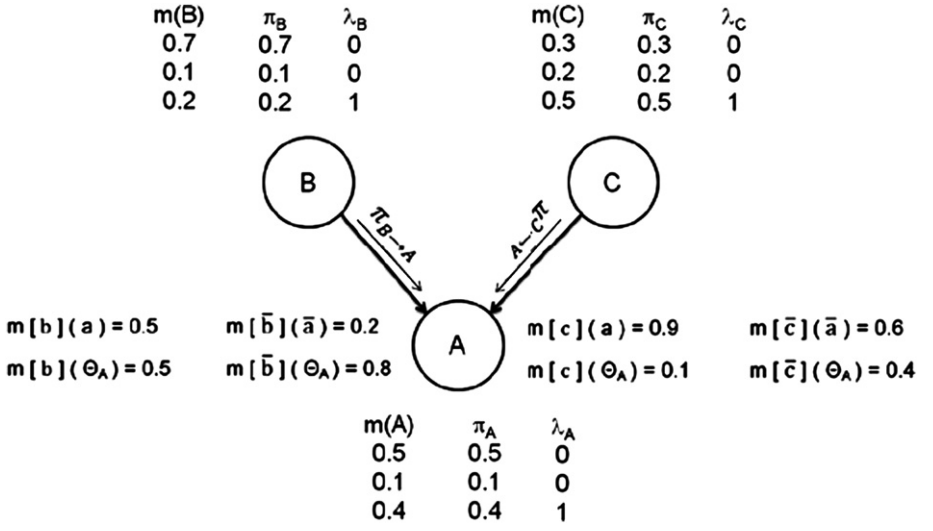


Fig. 3. The state of the simple directed evidential network after initialization.

Suppose now that a new observation O_A is introduced at a node A, where $O_A = (1, 0, 0)$. So, node A is instantiated, and for updating the belief network, the propagation algorithm described above will be performed.

Node A. A is instantiated (new observation O_A)

1. Compute the new values for the node A.
2. A has no children.
3. Compute and send $\lambda_{A\to B}$ to its parent B using Eq. (24). For any $b \subseteq \Theta_B$, $bel_{A\to B}(b)$ is given by its bba function:
   $m_{A\to B}(b) = 1 \times 0.2 = 0.2$,
   $m_{A\to B}(\bar{b}) = 0 \times 0.5$,
   $m_{A\to B}(\Theta_B) = 0.8$.
4. Similarly, compute and send $\lambda_{A\to C}$ to its parent C using Eq. (24).

Node B. When B receives a new λ message from A

1. Compute its new marginal using Eq. (21) (Table 3).
   After normalization, we obtain:

$$m^B(b) = \frac{0.74}{0.98} = 0.75, \quad m^B(\bar{b}) = 0.08, \quad m^B(\Theta_B) = 0.17,$$

where $k = 1 - 0.02 = 0.98$ is a normalization factor.

Table 3
Dempster's rule of combination for computing bel^B

	0.7	0.1	0.2
	b	$\bar{b}$	Θ_B
b	b	$\emptyset$	b
0.2	0.14	0.02	0.04
Θ_B	b	$\bar{b}$	Θ_B
0.8	0.56	0.08	0.16

Please cite this article in press as: B.B. Yaghlane, K. Mellouli, Inference in directed evidential networks based ..., Int. J. Approx. Reason. (2008), doi:10.1016/j.ijar.2008.01.002

UNCORRECTED P

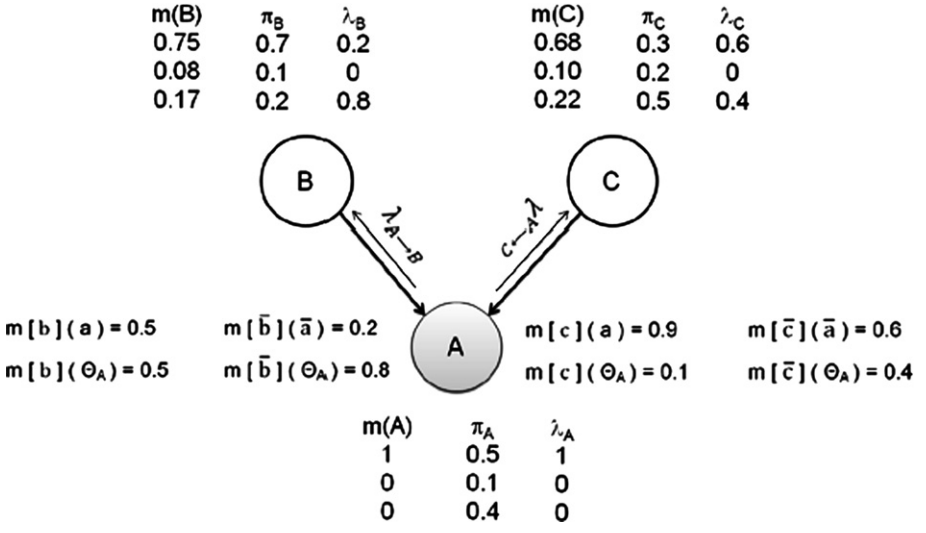


Fig. 4. The state of the belief network after A is instantiated.

2. The node B has no children (other than A) and no parent.

Node C. We do the same thing as node B.
The final state of the belief network is given by Fig. 4.

6. Propagation algorithm for multiply directed evidential networks

In this section, we propose a propagation algorithm for multiply directed evidential networks (DEVNs). The basic idea is to transform the initial DEVN into a binary join tree [23]. Some modifications for building the binary join tree are proposed. With these modifications, the graphical structure, called a (*modified*) *binary join tree* becomes clear, without ambiguity, and emphasizes explicitly the conditional relations. In order to avoid the computation of joint belief functions as required when using the fusion algorithm [22] in binary join trees, we use the DRC and the GBT for making inference efficiently in the (modified) binary join tree.

The rest of this section is organized as follows. In Section 6.1, we introduce the concept of binary join trees and its utility in reducing the number of combinations done in computing marginals. In Section 6.2, we explain how to transform an initial directed evidential network with conditional belief functions into a binary join tree. Finally, Section 6.3 presents an efficient algorithm applied to modified binary join trees based on the GBT and DRC and discusses some related computational aspects.

6.1. Binary join trees

A **join tree** consists of a set of nodes (or vertices) where each node is connected to one or more neighbor nodes and where the initially given potentials (or valuations) are distributed on the nodes. Marginals are then computed on the basis of a message-passing scheme, where nodes receive and send messages to their neighbor nodes. Formally, the definition of join tree is given as follows:

Definition 3 (*Join tree*). Let $\mathbf{G}=(\mathbf{N},\mathbf{E})$ be a tree where $\mathbf{N}$ is the set of nodes so that $N_i \in \mathbf{N}$ is a non-empty subset of $\mathbf{V} = \{X_1, \ldots, X_n\}$ and $\mathbf{E}$ is the set of edges (N_i, N_j) in $\mathbf{G}$. We say $\mathbf{G}$ is a *join tree* if for any two nodes N_i and N_j, $N_i \cap N_j \subseteq N_k$ where N_k is any node on the path between N_i and N_j.

In other words, $\mathbf{G}$ is a join tree if it satisfies the *Markov property*, it means that a variable which appears in two nodes appears also in every node on the intersection of the two nodes.

Please cite this article in press as: B.B. Yaghlane, K. Mellouli, Inference in directed evidential networks based ..., Int. J. Approx. Reason. (2008), doi:10.1016/j.ijar.2008.01.002

For two nodes N_i and N_j, we use $pth(N_i, N_j)$ to denote the set of nodes on the path between N_i and N_j. Since **G** is a tree, there is a unique path for every pair of nodes. In addition, the join tree must be completely connected, i.e. $pth(N_i, N_j) \neq \emptyset$ for all $1 \leqslant i, j \leqslant n$.

Nevertheless, the join tree is not very efficient when multiple marginals have to be computed due to the redundant combinations. To get rid of this drawback, Shenoy introduces the concept of *binary join tree* (BJT) which is a join tree according to which no node has more than three neighbors [23]. The basic idea behind a BJT is that all combinations are done in a binary basis.

In general, for a join tree, a node with m neighbor nodes would need m^2 combinations ($m(m-1)$ combinations for the computation of the messages and m combinations for the marginal). However, each join tree can be transformed into a binary join tree through addition of other nodes involving the reduction of the number of combinations. Then, for a node with m neighbor nodes, $3(m-1)$ combinations are needed ($3m-4$ combinations for the computation of all messages and one additional combination is required for the computation of the marginal). Without the use of a binary join tree, a lot of unnecessary combinations would be performed for nodes which have many neighbor nodes.

6.1.1. Binary join tree construction

Let us have a closer look at the binary join tree construction process using the chest clinic problem. Note that the formal procedure for constructing a binary join tree is described in [23].

Example 4. Chest clinic problem [14].

A doctor has to decide whether a patient, who complains about shortness-of-breath, suffers from bronchitis, lung cancer, or tuberculosis. The doctor's medical knowledge is summarized as follows: "Shortness-of-breath (dysponea) (***D***) may be due to either (***E***) tuberculosis (***T***) and lung cancer (***L***), or bronchitis (***B***). A recent visit to Asia (***A***) increases the chances of tuberculosis, while smoking (***S***) is known to be a risk factor for both lung cancer and bronchitis. The results of a single chest X-ray (***X***) do not discriminate between lung cancer and tuberculosis, nor does the presence or absence of dysponea."

In the join tree, we begin with a collection of potentials (or valuations) $\vartheta = \{\sigma, \tau, \lambda, \beta, \varepsilon, \xi, \delta, o_A, o_D\}$ that define the *joint distribution*. The domains of the potentials form a hypergraph $H = \{\{A\}, \{S\}, \{A,T\}, \{S,L\}, \{S,B\}, \{T,L,E\}, \{E,X\}, \{E,B,D\}\}$. The BJT construction process follows these steps:

- *Arrange the subsets of H in a binary join tree* so that no node has more than three neighbors. A binary join tree for the chest clinic problem is given in Fig. 5. It is important to have all singleton subsets in a binary join tree.
- *Attach singleton subsets to the binary join tree*. This can be done in different manners. In [15], we find a method for minimizing the number of computations required to compute the marginal for that singleton subset.

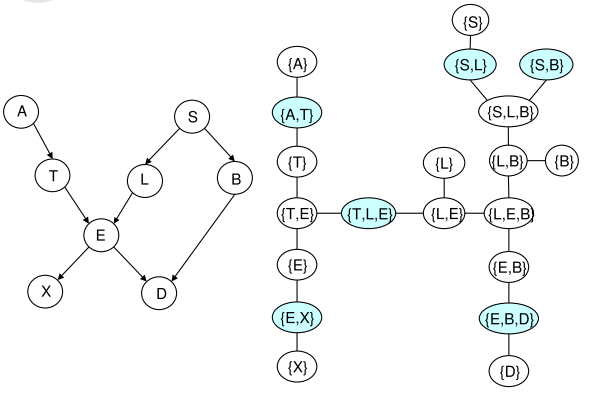


Fig. 5. The DAG and the corresponding binary join tree for chest clinic problem.

Please cite this article in press as: B.B. Yaghlane, K. Mellouli, Inference in directed evidential networks based ..., Int. J. Approx. Reason. (2008), doi:10.1016/j.ijar.2008.01.002

- *Make the tree binary again* if the tree becomes non-binary when attaching a singleton subsets to it. This can be done by duplicating the node that has four neighbors, distributing the four neighbors equally between the two copies of node and drawing an edge between the two duplicated nodes.

6.1.2. Computing marginals using the fusion algorithm

In this section, we briefly describe the efficient process of computing marginals using the fusion algorithm. Suppose that we have a set of valuations ϑ, the combination of all valuations in ϑ is called the *joint valuation*, i.e. $\phi = \oplus\{\vartheta_i | i = 1, \ldots, m\}$. Our problem is to find the marginal of the joint valuation for some variables or subsets of the variables S, i.e. $\phi^{\downarrow S}$. However, when there are many variables to manipulate, this method is not feasible for making inference since the joint frame of the variables is too large. To solve this problem, many researchers have studied methods for computing marginals without computing the joint valuation. A well-known approach is the local computation. Shenoy [23] described the phases of this approach as follows:

1. *Binary join tree construction phase.* Transform the original directed network to binary join tree and add singleton subsets for which we need marginals to the binary join tree (if these singleton subsets are not already included). Then, associate each valuation in ϑ with the corresponding node.
2. *Transfer of valuations phase.* Transfer the valuations from the leaves to the adjacent supersets in order to avoid unnecessary computations.
3. *Message-passing phase.* Apply the fusion algorithm using local computations.
4. *Computing marginals phase.* Compute the marginals for the desired subsets.

When we focus on the belief functions theory, we remark that since the calculations of the marginals are done with joint belief functions on the product space of the involved variables, it requests more storage space and more computations. In addition, it has for a long time been the reason why the inference architecture based on the secondary computational structure do not really use the direction on the edges in the original network (directed evidential network). In fact, a careful exploitation of the independence relations would result in less complex structure. In the following section, we discuss this problem on belief functions formalism.

6.2. From directed evidential networks to modified binary join trees

In this section, we present a computational data structure based on the binary join tree and maintaining the (in)dependence relations of the original directed evidential network (DEVN) with conditional belief functions. By exploiting the structure of the original directed evidential network, we make some modifications on the obtained binary join tree. These modifications explicitly show the independence relations.

6.2.1. Modified binary join tree

It is important to realize that binary join trees represent a tradeoff between computing time and memory space. By storing intermediate results, less computing time is needed for the sake of using more memory space.

Special care has to be taken for nodes having a large number of neighbor nodes. For such nodes it happens that the combination of incoming messages generate huge mass functions even if the incoming messages were relatively small. Nevertheless, it is still possible that every outgoing message is relatively small because a marginalization was performed. In a binary join tree, these huge mass functions representing intermediate results would have been stored and therefore much memory space would have been lost.

Nevertheless, when we transform the original directed evidential network into a binary join tree, we lose some useful information about the relationships between the variables. For instance, when we have the node TLE in the binary join tree corresponding to the chest clinic problem (Fig. 5), we cannot deduce explicitly that this node represents the conditional belief over E given its parents T and L, denoted by $E|TL$.

In order to avoid this drawback, we suggest the following modifications to the evidential binary join tree. We propose to draw a *rectangle* containing the conditional relations between variables (like $X|Y$) instead of a circle containing just the list of these variables. With this minor modification, the obtained graphical structure is no longer ambiguous and emphasizes explicitly the conditional relations (see Fig. 6).

UNCORRECTED PROOF

Please cite this article in press as: B.B. Yaghlane, K. Mellouli, Inference in directed evidential networks based ..., Int. J. Approx. Reason. (2008), doi:10.1016/j.ijar.2008.01.002

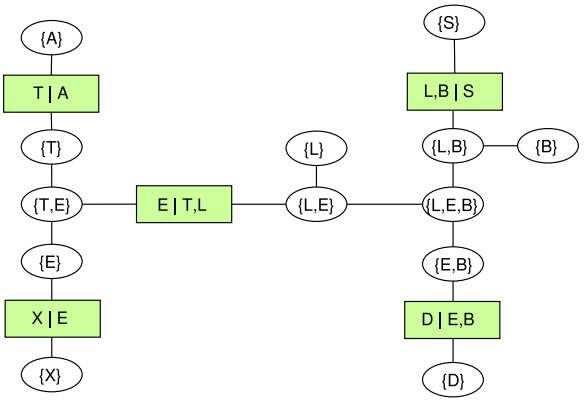


Fig. 6. The (modified) binary join tree for chest clinic problem.

6.2.2. Getting joint beliefs

If we want the joint belief function on the frame $\Theta_X \times \Theta_Y$ and these two variables are not present in a node of the tree, we can obtain it by adding a new variable called Z whose parents are X and Y and with a conditional belief function on Z given XY that represents the logical relation.

Let $X = \{x_i : i = 1, \ldots, n_X\}$ and $Y = \{y_j : j = 1, \ldots, n_Y\}$.
Define $Z = \{z_{ij} : i = 1, \ldots, n_X, j = 1, \ldots, n_Y\}$ where $z_{ij} = (x_i, y_j)$.
Define the set of conditional bba's $\{m^Z[x_i, y_j], i = 1, \ldots, n_X, j = 1, \ldots, n_Y\}$with

$$\{m^Z[x_i, y_j](z_{ij}) = 1, i = 1, \ldots, n_X, j = 1, \ldots, n_Y\}.$$

The belief function collected on Z after the propagation phase is equal to the requested belief function on the joint frame $\Theta_X \times \Theta_Y$. Generalization to any set of variables is immediate.

6.3. Efficient computation using DRC and GBT in the modified binary join tree

In the following, we will first explain, in Section 6.3.1, how to perform the propagation with conditional beliefs profiting of the original structure of the network. Then, in Section 6.3.2, we will discuss some computational aspects related to our algorithm.

6.3.1. Propagation with conditional beliefs

The belief functions presented in the directed evidential network are in the conditional form [1]. This solution seems to be faster than the general one (i.e. joint form) as developed by Shenoy in binary join tree [23] and which does not profit from the particular nature of the conditional belief functions.

Consider the simple case where in the (modified) binary join tree we have the next two quite common propagation patterns. The propagation process is then performed by using the GBT and the DRC (Fig. 7).

6.3.1.1. Propagation in the GBT case. Let $\{m^X[y_i] : y_i \in \Theta_Y\}$ be the set of conditional bba's at node $X|Y$. Suppose m_0^X is the bba collected at node X that must be propagated through node $X|Y$ in order to compute m_1^Y. Using the GBT, we get:

$$\begin{aligned} b_1^Y(y) &= \sum_{x \subseteq \Theta_X} m_0^X(x) \prod_{y_i \in \bar{y}} b^X[y_i](\bar{x}), \\ pl_1^Y(y) &= \sum_{x \subseteq \Theta_X} m_0^X(x) \left(1 - \prod_{y_i \in y} (1 - pl^X[y_i](x))\right). \end{aligned} \tag{25}$$

Please cite this article in press as: B.B. Yaghlane, K. Mellouli, Inference in directed evidential networks based ..., Int. J. Approx. Reason. (2008), doi:10.1016/j.ijar.2008.01.002

UNCORRECTED PR

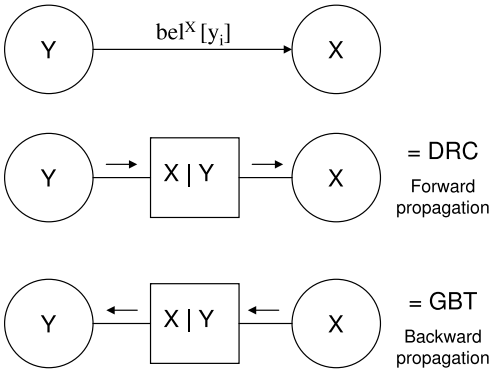


Fig. 7. GBT and DRC propagation patterns.

Table 4 presents an example of GBT propagation with $X = \{x_1, x_2\}$ and $Y = \{y_1, y_2, y_3\}$. The initial vectors are m^X and $pl^X[y_i] : i = 1, 2, 3$. The plausibilities for $|y| \geqslant 2$ are computed with $1 - \prod_{y_i \in y}(1 - pl^X[y_i])$. The line $pl^Y[m^X]$ is computed with relation (25).

6.3.1.2. Propagation in the DRC case. Let $\{m^X[y_i] : y_i \in \Theta_Y\}$ be the set of conditional bba's at node $X|Y$. Suppose m_0^Y is the bba collected at node Y that must be propagated through node $X|Y$ in order to compute m_1^X. Using the DRC, we get:

$$b_1^X(x) = \sum_{y \subseteq Y} m_0^Y(y) \prod_{y_i \in y} b^X[y_i](x),$$
$$pl_1^X(x) = \sum_{y \subseteq Y} m_0^Y(y) \left(1 - \prod_{y_i \in y}(1 - pl^X[y_i](x))\right). \quad (26)$$

Table 5 presents an example of DRC propagation with the same values as in Table 4. The initial vectors are m^Y and $pl^X[y_i] : i = 1, 2, 3$. The column $pl^X[m^Y]$ is computed with relation (26).

Table 4
Example of GBT propagation

2^X	m^X	$pl^X[\cdots]$						
		$[y_1]$	$[y_2]$	$[y_3]$	$[y_1, y_2]$	$[y_1, y_3]$	$[y_2, y_3]$	$[Y]$
$\emptyset$	0	0	0	0	0	0	0	0
x_1	0.3	1	0.7	0.4	1	1	0.82	1
x_2	0.5	0.9	0.9	1	0.99	1	1	1
X	0.2	1	0.9	1	1	1	1	1
$pl^Y[m^X]$		0.95	0.84	0.82	0.99	1	0.95	1

Table 5
Example of DRC propagation

2^X	$pl^X[\cdots]$							$pl^X[m^Y]$
	$[y_1]$	$[y_2]$	$[y_3]$	$[y_1, y_2]$	$[y_1, y_3]$	$[y_2, y_3]$	$[Y]$	
$\emptyset$	0	0	0	0	0	0	0	0
x_1	1	0.7	0.4	1	1	0.82	1	0.86
x_2	0.9	0.9	1	0.99	1	1	1	0.97
X	1	0.9	1	1	1	1	1	1
m^Y	0.3	0	0.2	0	0	0.1	0.4	

UNCORRECTED PROOF

Please cite this article in press as: B.B. Yaghlane, K. Mellouli, Inference in directed evidential networks based ..., Int. J. Approx. Reason. (2008), doi:10.1016/j.ijar.2008.01.002

6.3.2. Computational aspects

Let $\mathbf{U} = \{X_1, \ldots, X_n\}$ be a finite set of variables where each X_i has its frame Θ_{X_i}. Let X and Y be two subsets of $\mathbf{U}$. According to the notation of the previous section, a *conditional belief function* for X given Y is represented by $bel^X[y]$ where $y \in \Theta_Y$. Similarly, the *joint belief functions* are defined on the space $\Theta_X \times \Theta_Y$.

At node $X|Y$, the memory requirement is at worst $|\Theta_Y| \times 2^{|\Theta_X|}$, and not $2^{|\Theta_X|\times|\Theta_Y|}$ as in general case. The reduction can be very large. In our example presented in Tables 4 and 5, we store $12 = 3 \times 2^2$ values, and not $64 = 2^{3\times 2}$.

Furthermore, in both cases (i.e. GBT and DRC), there is no need to compute the belief function over the $\Theta_X \times \Theta_Y$ space, except if one specifically asks for it. This reduces enormously the computational load. To store all conditional bba's, it needs only $|\Theta_Y| \times 2^{|\Theta_X|}$ in the worst case. In practice it should be about $|\Theta_Y| \times |\Theta_X|$ data if one stores only the positive masses, and one assumes that the conditional bba's are of the same complexity that a probability counterpart (often it will even be less). A conditioning for a given $[y_i]$ costs about $|\Theta_X|$ additions, to be done $|\Theta_Y|$ times for each focal elements of m_0. So the total number of operations is about $|\Theta_X| \times |\Theta_Y| \times |\Theta_Y| \times |\Theta_X| = (|\Theta_Y| \times |\Theta_X|)^2$ to get $m_0^{XY} \textcircled{\cap} m_1^{XY}$.

7. Conclusion

In this paper, we have discussed some aspects related to uncertainty representation and reasoning in directed evidential networks with conditional belief functions. Uncertainty is expressed by belief functions as understood in the context of the transferable model (TBM). Directed evidential networks with conditional belief functions have the advantage of providing an appropriate representation of the knowledge that can be produced as conditional relationships.

On the one hand, the propagation algorithms for these networks are proposed for reasoning in singly connected directed evidential networks and in multi-connected directed evidential networks. Furthermore, we have shown how to do efficient computations using GBT and DRC rules proposed by Smets. These rules make possible the representation of knowledge by conditional belief functions and then the reasoning directly in the DEVN. When the network is multi-connected, we have proposed a computational data structure based on the binary join tree and maintaining the independence relations of the original directed evidential network with conditional belief functions. This solution seems to be faster than the general one (i.e., the joint form) as developed by Shenoy in binary join tree which does not profit from the particular nature of the belief functions.

On the other hand, we think that it is important to look at approximation methods [30] because exact numerical computation is not possible for some examples. In any case, an approximated numerical result which is close to the exact value may often be sufficient for the user. One of the advantages of this method is that it computes besides the approximated result also an interval as a bound of the exact value [9,10]. In addition, the mixture of exact and approximate methods remains a topic for future research.

Acknowledgements

The authors are grateful to the anonymous reviewers for their numerous suggestions and constructive comments on various aspects of this paper. Special thanks to Mohsen Antit for his help.

References

[1] B. Ben Yaghlane, K. Mellouli, Belief function propagation in directed evidential networks, in: Proceedings of International Conference on Information Processing and Management of Uncertainty in Knowledge Based Systems (IPMU'06), 2006, pp. 1451–1458.
[2] B. Ben Yaghlane, Ph. Smets, K. Mellouli, Belief function independence: I. The marginal case, International Journal of Approximate Reasoning 29 (1) (2002) 47–70.
[3] B. Ben Yaghlane, Ph. Smets, K. Mellouli, Belief function independence: II. The conditional case, International Journal of Approximate Reasoning 31 (1–2) (2002) 31–75.
[4] B. Ben Yaghlane, Ph. Smets, K. Mellouli, Directed evidential networks with conditional belief functions, in: T.D. Nielsen, N.L. Zhang (Eds.), Proceedings of ECSQARU-2003, LNAI 2711, Springer-Verlag, 2003, pp. 291–305.
[5] J. Cano, M. Delgado, M. Serafin, An axiomatic framework for propagating uncertainty in directed acyclic networks, International Journal of Approximate Reasoning 8 (1993) 253–280.

Please cite this article in press as: B.B. Yaghlane, K. Mellouli, Inference in directed evidential networks based ..., Int. J. Approx. Reason. (2008), doi:10.1016/j.ijar.2008.01.002

UNCORRECTED PROOF

[6] B.R. Cobb, P.P. Shenoy, On the plausibility transformation method for translating belief function models to probability models, International Journal of Approximate Reasoning 41 (2006) 314–330.
[7] L.M. de Campos, Independency relationships and learning algorithms for singly connected networks, Journal of Experimental and Theoretical Artificial Intelligence 10 (1998) 511–549.
[8] A. Dempster, Upper and lower probabilities induced by a multivalued mapping, Annals of Mathematical Statistics 38 (1967) 325–339.
[9] T. Denoeux, Inner and outer approximation of belief structure using a hierarchical clustering approach, International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 9 (4) (2001) 437–460.
[10] R. Haenni, N. Lehmann, Resource bounded and anytime approximation of belief function computations, International Journal of Approximate Reasoning 31 (2002) 103–154.
[11] R. Haenni, J. Kohlas, N. Lehmann, Probabilistic argumentation systems, in: Handbook of Defeasible Reasoning and Uncertainty Management Systems, in: J. Kohlas, S. Moral (Eds.), Algorithms for Uncertainty and Defeasible Reasoning, Vol. 5, Kluwer Academic Publishers, Dordrecht, 2001, pp. 221–288.
[12] J. Kim, J. Pearl, A computational modal for causal and diagnostic reasoning in inference engines, in: International Joint Conference on Artificial Intelligence IJCAI'83, Morgan Kaufman, San Mateo, CA, 1983, pp. 190–193.
[13] J. Kohlas, P.A. Monney, A Mathematical Theory of Hints: An Approach to Dempster–Shafer Theory of Evidence, Lecture Notes in Economics and Mathematical Systems, vol. 425, Springer-Verlag, 1995.
[14] S.L. Lauritzen, D.J. Spiegelhalter, Local computation with probabilities and graphical structures and their application to expert systems, Journal Royal Statistical Society B 50 (1988) 157–224.
[15] V. Lepar, P.P. Shenoy, A comparison of Lauritzen–Spiegelhalter, Hugin, and Shenoy–Shafer architectures for computing marginals of probability distributions, in: Proceedings of Uncertainty in Artificial Intelligence (UAI'98), 1998, pp. 328–337.
[16] J. Pearl, Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference, Morgan Kaufmman, Los Altos, CA, 1988.
[17] Ch. Simon, Ph. Weber, Bayesian Networks Implementation of the Dempster Shafer Theory to Model Reliability Uncertainty, in: ARES'06, 2006, pp. 788–793.
[18] G. Shafer, Mathematical Theory of Evidence, Princeton University Press, Princeton, 1976.
[19] G. Shafer, P.P. Shenoy, K. Mellouli, Propagating belief functions in qualitative Markov trees, International Journal of Approximate Reasoning 1 (4) (1987) 349–400.
[20] P.P. Shenoy, Valuation-based systems: a framework for managing uncertainty in expert systems, in: L.A. Zadeh, J. Kacprzyk (Eds.), Fuzzy Logic for the Management of Uncertainty, John Wiley and Sons, New York, 1992, pp. 83–104.
[21] P.P. Shenoy, Valuation networks and conditional independence, in: D. Heckerman, A. Mamdani (Eds.), Proceedings of Uncertainty in Artificial Intelligence (UAI'93), Morgan Kaufmann, San Mateo, CA, USA, 1993, pp. 191–199.
[22] P.P. Shenoy, A new method for representing and solving Bayesian decision problems, in: D.J. Hand (Ed.), Artificial Intelligence Frontiers in Statistics: AI and Statistics III, Chapman and Hall, London, 1993, pp. 119–138.
[23] P.P. Shenoy, Binary join trees for computing marginals in the Shenoy–Shafer architecture, International Journal of Approximate Reasoning 17 (1997) 239–263.
[24] P.P. Shenoy, G. Shafer, Propagating belief functions with local computations, IEEE Expert 1 (3) (1986) 43–52.
[25] P.P. Shenoy, G. Shafer, Axioms for probability and belief functions propagation, in: Proceedings of Uncertainty in Artificial Intelligence (UAI'90), 1990, pp. 159–198.
[26] Ph. Smets, Un modèle mathématico-statistique simulant le processus du diagnostic médical, Ph.D Thesis, Université Libre de Bruxelles, 1978 (available through University Microfilm International, 30-32 Mortimer street, London W1N 7RA, thesis 80-70,003).
[27] Ph. Smets, Belief functions: the disjunctive rule of combination and the generalized Bayesian theorem, International Journal of Approximate Reasoning 9 (1993) 1–35.
[28] Ph. Smets, R. Kennes, The transferable belief model, Artificial Intelligence 66 (1994) 191–234.
[29] P. Walley, Statistical Reasoning with Imprecise Probabilities, Chapman and Hall, London, 1991.
[30] N. Wilson, Algorithms for Dempster–Shafer theory, in: D.M. Gabbay, Ph. Smets (Eds.), Handbook of Defeasible Reasoning and Uncertainty Management Systems, vol. 5, Kluwer, The Netherlands, 2000, pp. 421–475.
[31] H. Xu, Uncertainty Reasoning and Decision Analysis using Belief Functions in the Valuation-Based Systems, Ph.D. Thesis, Université Libre de Bruxelles, 1995.
[32] H. Xu, Ph. Smets, Evidential reasoning with conditional belief functions, in: D. Heckerman et al. (Eds.), Proceedings of Uncertainty in Artificial Intelligence (UAI'94), Morgan Kaufmann, San Mateo, CA, 1994, pp. 598–606.

UNCORRECTED PROOF

Please cite this article in press as: B.B. Yaghlane, K. Mellouli, Inference in directed evidential networks based ..., Int. J. Approx. Reason. (2008), doi:10.1016/j.ijar.2008.01.002