# Compiling Min-based Possibilistic Causal Networks: A mutilated-based Approach

Raouia Ayachi[1,2], Nahla Ben Amor[1], and Salem Benferhat[2]

[1] LARODEC, Institut Supérieur de Gestion Tunis, Le Bardo, Tunisie, 2000.
raouia.ayachi@gmail.com,nahla.benamor@gmx.fr
[2] CRIL-CNRS, Université d'Artois, France, 62307
benferhat@cril.univ-artois.fr

**Abstract.** Qualitative causal possibilistic networks are important tools for handling uncertain information in the possibility theory framework. Despite their importance, no compilation has been performed to ensure causal reasoning in possibility theory framework. This paper proposes two compilation-based inference algorithms for min-based possibilistic causal networks. The first is a possibilistic adaptation of the probabilistic inference method [8] and the second is a purely possibilistic approach. Both of them are based on an encoding of the network into a propositional theory and a compilation of this output in order to efficiently compute the effect of both observations and interventions, while adopting a mutilation strategy.

## 1 Introduction

Possibilistic networks [2] provide efficient tools to deal with uncertain data. They compactly represent the prior background knowledge and efficiently reason in the presence of new information. While the quantitative (or product-based) possibilistic networks are very similar to probabilistic Bayesian networks, the qualitative (or min-based) ones, which are the focus of this paper, have significant differences. Emphasis has recently placed on inference in possibilistic networks [2, 4], especially when it is dealt with *compilation* [1].

Causal possibilistic networks are updated in the presence of two types of information: *observations* which are the results of testing some variables and *interventions* which correspond to external actions forcing some variables to have some specific values. From a representational point of view, interventions are distinguished from observations using the concept of the 'do' operator [13, 15]. From a reasoning point of view, an intervention on a variable $A$ is represented using the so-called mutilation, by ignoring relations between the intervened variable $A$ and its direct causes. Handling sets of observations and interventions is an important issue that can be useful where some variables are directly observed and/or forced to take some values by performing interventions.

In [1], we have proposed compilation-based inference methods for min-based possibilistic networks that only deal with observations. The idea in [1] consists in encoding the network using a propositional theory and then compiling the resulting encoding to have a polytime possibilistic inference. However, there is no compilation that has been proposed for min-based possibilistic causal networks that takes into account the concept of interventions. In this paper, we will propose two mutilated-based approaches which deal with inference in min-based possibilistic causal networks under a compilation framework. Our objective is to ensure an efficient computation of the effect of both observations and interventions by avoiding a re-compilation of the network each time an intervention or an observation is taken place, which is considered intractable.

The remaining paper is organized as follows: Section 2 presents a brief refresher on possibility theory and compilation languages. Section 3 describes the inference process using the so-called mutilated $\Pi$-DNNFs (*Possibilistic Decomposable Negation Normal Form*). Inference using mutilated compiled possibilistic bases is presented in Section 4. Section 5 concludes the paper.

## 2 Basic Backgrounds on Possibility and Compilation

Let $V = \{X_1, X_2, ..., X_N\}$ be a set of variables. We denote by $D_{X_i} = \{x_1, .., x_n\}$ the domain associated with the variable $X_i$. By $x_i$ we denote any instance of $X_i$. By $x_{ij}$ we denote the $j^{th}$ instance of $X_i$. When there is no confusion we use $x_i$ to mean any instance of $X_i$. $\Omega$ denotes the universe of discourse, which is the Cartesian product of all variable domains in $V$. Each element $\omega \in \Omega$ is called a state of $\Omega$. $\omega[X_i] = x_i$ denotes an instantiation of $X_i$ in $\omega$.

### 2.1 Possibility Theory

One of the basic concepts in possibility theory (see [11] for more details) is the concept of possibility distribution, denoted by $\pi$. It is a mapping from $\Omega$ to the unit interval [0, 1]. In this paper, we consider the qualitative interpretation of this scale where only the ordering induced by degrees is important. Given a possibility distribution $\pi$, we can define a mapping grading the possibility measure of an event $\phi \subseteq \Omega$ by $\Pi(\phi) = max_{\omega \in \phi} \pi(\omega)$. $\Pi$ has a dual measure which is the necessity measure defined by $N(\phi) = 1 - \Pi(\neg\phi)$.

Conditioning consists in modifying our initial knowledge, encoded by $\pi$, by the arrival of a new certain piece of information $\phi \subseteq \Omega$. The qualitative setting leads to the well known definition of min-conditioning [11]:

$$\Pi(\psi \mid \phi) = \begin{cases} \Pi(\psi \wedge \phi) \ if \ \Pi(\psi \wedge \phi) < \Pi(\phi) \\ 1 \qquad\qquad\quad \text{otherwise.} \end{cases} \tag{1}$$

One of the well-used and developed compact representations of a possibility distribution is the concept of a possibilistic knowledge base [14]. Denoted by $\Sigma$, it is made up of a finite set of weighted formulas. Formally,

$$\Sigma = \{(\alpha_i, a_i), i = 1, .., n, a_i \neq 0\}. \tag{2}$$

Each possibilistic logic formula $(\alpha_i, a_i)$ expresses that the propositional formula $\alpha_i$ is certain to at least the level $a_i$, or more formally by $N(\alpha_i) \geq a_i$, where $N$ is the necessity measure associated to $\alpha_i$.

The following subsection represents another compact representation of possibility distribution that deals with both observations and interventions.

### 2.2 Possibilistic Causal Networks

A possibilistic causal network is a graphical way to represent uncertain information [4]. Over a set of variables $V$, a possibilistic causal network, denoted by $\Pi G_{min}$ is composed of:
- A *graphical component* that is a DAG where nodes represent variables and edges encode not only dependencies between variables but also direct causal relationships [4]. The parent set of any variable $X_i$, denoted by $U_i = \{U_{i1}, U_{i2}, ..., U_{im}\}$ where $m$ is the number of parents of $X_i$, represents all direct causes for $X_i$. In what follows, we use $x_i$, $u_i$, $u_{ij}$ to denote, respectively, possible instances of $X_i$, $U_i$ and $U_{ij}$.
- A *numerical component* that quantifies different links. Uncertainty of each node in $\Pi G_{min}$ is represented by a local normalized possibility distribution in the context of its parents (i.e., $\forall u_i, max_{x_i} \Pi(x_i|u_i) = 1$).

The set of a priori and conditional possibility degrees in a $\Pi G_{min}$ induces a unique joint possibility distribution defined by the following min-based chain rule:

$$\pi(X_1, .., X_N) = \min_{i=1..N} \quad \Pi(X_i \mid U_i). \tag{3}$$

Causal networks are updated in the presence of two types of information: set of *observations* (evidences) which are the results of testing some variables, and a set of *interventions* which represent external events, coming from outside the system and forcing some variables to take some specific values [15]. Interventions, denoted by $do(x_I)$ have a reasoning and a representational interpretations. This paper focuses on the reasoning aspect.

**Mutilation** From a reasoning point of view, an intervention is handled by the so-called *mutilation* operation [15], which refers to altering the network structure by excluding all direct causes related to the variable of interest and maintaining the remaining variables intact [15]. The possibility distribution associated with the mutilated network $\Pi G_{mut}$ is denoted by $\pi_m$. The effect of $do(x_I)$ is to transform $\pi(\omega)$ into $\pi_m(\omega|x_I)$, which gives us [4]:

$$\forall \omega; \pi_m(\omega|x_I) = \pi(\omega|do(x_I)). \tag{4}$$

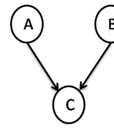
By mutilating the network, parents of $X_I$ become independent of $X_I$. Moreover, the event that attributes the value $x_I$ to $X_I$ becomes sure after performing intervention $do(x_I)$. More formally, $\pi_m(x_I) = 1$ and $\forall x_i, x_i \neq x_I, \pi_m(x_i) = 0$. The effect of $do(x_I)$ on $\pi$ is given as follows, $\forall \omega$:

$$\pi(\omega|do(x_I)) = \begin{cases} min_{i \neq I} \quad \pi(x_i|u_i) \; if \; \omega[X_i] = x_I \\ 0 \qquad\qquad\qquad\quad \text{otherwise.} \end{cases} \tag{5}$$

**Example 1** *Let us consider the $\Pi G_{min}$ of Figure 1.*

*Let $C$ be the variable in $\Pi G_{min}$ forced to take the value $c_1$ by the intervention $do(c_1)$. The possibility distribution $\pi_m(A, B, C)$ associated with $\Pi G_{mut}$ represents the effect of $do(c_1)$ on $\pi(A, B, C)$. The intervention $do(c_1)$ implies $\pi_m(c_1) = 1$ and $\pi_m(c_2) = 0$. For instance, $\pi(a_1, b_2, c_1|do(c_1)) = \pi_m(a_1, b_2, c_1) = min(\pi_m(a_1), \pi_m(b_2), \pi_m(c_1)) = min(0.7, 0.4, 1) = 0.4$.*

| | Π(C\|A,B) | |
|---|---|---|
| | c1 | c2 |
| a1,b1 | 1 | 0.3 |
| a1,b2 | 0.1 | 1 |
| a2,b1 | 0.5 | 1 |
| a2,b2 | 0.8 | 1 |




| Π(A) | |
|---|---|
| a1 | a2 |
| 0.7 | 1 |

| Π(B) | |
|---|---|
| b1 | b2 |
| 1 | 0.4 |

**Fig. 1.** A causal possibilistic network $\Pi G_{min}$.

The effect of interventions on the remaining network is defined by applying conditioning on the mutilated network after observation as follows:

**Proposition 1** *Let $\Pi G_{min}$ be a min-based possibilistic causal network. Let $do(x_I)$ be an intervention forcing $X_I$ to take the value $x_I$. Let $\Pi G_{mut}$ be the mutilated network obtained after mutilation. Then, $\forall \omega, \forall x_I \in D_{X_I}$,$\pi(\omega|do(x_I))=$ $\pi_m(\omega|X_I = x_I)$.*

### 2.3 Compilation concepts

Knowledge compilation is an artificial intelligence area related to a mapping problem from intractable logical theories (typically, from propositional knowledge bases in a CNF form) into suitable target compilation languages. These latters are characterized by a *succinctness* criteria and a set of *queries* and *transformations* performed in polynomial time with respect to the size of compiled bases [6]. There are several compilation languages as it has been studied in the knowledge map of [10]. We are in particular interested in *Decomposable Negation Normal Form (DNNF)* [7] and *Valued Negation Normal Form (VNNF)* [12].

**DNNF language** The *Negation Normal Form (NNF)* language represents the pivotal language from which a variety of target compilation languages give rise by imposing some conditions on it. For instance, the DNNF language is the set of all NNFs satisfying *decomposability*: conjuncts of any conjunction share no variables [7]. DNNF supports a rich set of polynomial-time operations which can be performed simply and efficiently. Our choice is especially motivated by the set of operations that supports and its succinctness [10].

DNNF supports several transformations and queries. We restrict our attention to conditioning and forgetting operations:

- *Conditioning*: Let $\alpha$ be a propositional formula. Let $\rho$ be a consistent term, then conditioning $\alpha$ on $\rho$, denoted by $\alpha|\rho$ generates a new formula in which each propositional variable $P_i \in \alpha$ is set to:

$$P_i = \begin{cases} \top \text{ if } \; P_i \text{ is consistent with } \rho \text{ [3]} \\ \bot \; \text{ otherwise }. \end{cases} \tag{6}$$

[3] $P_i$ is consistent with $\rho$ if there exists an interpretation that satisfies both $P_i$ and $\rho$.

- *Forgetting*: Let $\alpha$ be a propositional formula. Let $P$ be a finite set of propositional variables $P_i$, then the forgetting of $P$ from $\alpha$, denoted by $\exists P.\alpha$ is equivalent to a formula that does not mention any variable $P_i$ from $P$. It can be inductively defined as follows:

$$\exists P_i.\alpha = \alpha|P_i \vee \alpha|\neg P_i. \tag{7}$$

where $\alpha|P_i$ (resp. $\alpha|\neg P_i$) is the result of conditioning of $\alpha$ on $P_i$ (resp. $\neg P_i$).

**VNNF language** All subsets of NNFs, known as valuable representation languages for boolean functions, have been extended to represent an enriched class of functions ranging over an ordered scale, namely *Valued Negation Normal Forms (VNNFs)* [12]. The VNNF language is fully characterized by a *representation context* $\prec \varepsilon, Y, F \succ$ consisting of a *valuation structure* $\varepsilon$, a finite set $Y$ of variables ranging on finite domains and a set $F$ of primitive or local functions, i.e., functions representing preferences or plausibility degrees over assignments. By valuation structure, we mean a triple $\varepsilon = \prec E, \geq, OP \succ$ where $(E, \geq)$ is a set ordered by a relation $\geq$ and $OP$ is the set of all binary operators $\otimes$ on $E$. $OP$ may contain the operators $\vee$ and $\wedge$. When $\geq$ is a total order, min and max are alternative generalizations of the boolean connectives [12]. The VNNF framework supports a larger family of queries, such as *optimization*, etc. It also supports several transformations, namely $\otimes$-*variable elimination* (a generalization of classical forgetting by using $\otimes$ instead of $\vee$ in equation (7)).

$\Pi$-DNNF [1] is a possibilistic version of DNNF in which conjunctions and disjunctions are substituted by minimum and maximum operators, respectively. It is considered as a special case of VNNFs [12] in which $E = [0, 1]$ and $OP$ is restricted to min and max operators.

## 3 Causal Inference using $\Pi$-DNNFs

In [1], we already proposed a possibilistic adaptation of the so-called arithmetic circuit method [8]. This adaptation requires the use of the $\Pi$-DNNF language [1] instead of the propositional DNNF language [7]. The main idea is based on encoding the possibilistic network using the CNF propositional language, then compiling it to infer in polytime (i.e., compute efficiently a posteriori possibility degrees given some evidence on a set of variables). The CNF encoding that we have used takes advantage of the structure exhibited by network parameters, known as *local structure*, which induces a reduction of the time and the size of factorization [9]. The question is whether this encoding can be adapted to deal with both observations and interventions. This section shows that the answer is yes. Of course, since we offer more flexibility, there is an extra-cost. In fact, handling interventions requires that a unique variable should be assigned to each parameter, while when we only deal with observations, different parameters (degrees) may be encoded by the same propositional variable. This means that local structure is only allowed in the non-intervention strategy.

### 3.1 Did we first mutilate the network?

One simple way for handling sets of interventions consists in mutilating $\Pi G_{min}$, encoding the resulting graph using a propositional theory and compiling it to offer a polynomial-time handling of queries. But, handling sets of interventions by this way is not efficient since it requires a re-compilation of the network each time an intervention is obtained, which is intractable. Our main contribution consists in allowing the treatment of both observations and interventions by avoiding the re-compilation of the network in case of sequences of observations or interventions are taken place. Handling interventions by mutilation is considered worthwhile since the initial network is vanished after mutilation, while we focus on computing the effect of both observations and interventions. For this reason, in the following subsection we will exhibit the appropriate trick that allows us to ensure such computation using only one compilation step.

### 3.2 Inference Process

Given a $\Pi G_{min}$, we should first encode it using the CNF representation language. Using two types of propositional variables namely, *evidence indicators* $\lambda_{x_i}$ for recording evidences and *network parameters* $\theta_{x_i|u_i}$ for recording possibility degrees, the CNF encoding is defined as follows [1]:

**Definition 1** *Let $\Pi G_{min}$ be a possibilistic causal network,* $\lambda_{x_{ij}}, (i = 1, .., N), (j = 1, .., n)$ *be the set of evidence indicators and* $\theta_{x_i|u_i}$ *be the set of parameter variables, then the encoding* $C_{min}$ *should contain the following clauses:*

– $\forall\ X_i \in V, C_{min}$ *contains the following two clauses (named indicator clauses):*

$$\lambda_{x_{i1}} \vee \lambda_{x_{i2}} \vee \cdots \lambda_{x_{in}} \tag{8}$$

$$\neg\lambda_{x_{ij}} \vee \neg\lambda_{x_{ik}}, j \neq k \tag{9}$$

– $\forall\ \theta_{x_i|u_i}$ *s.t* $u_i = \{u_{i1}, u_{i2}, ..., u_{im}\}$, $C_{min}$ *contains the following clauses:*

$$\lambda_{x_i} \wedge \lambda_{u_{i1}} \wedge \ldots \wedge \lambda_{u_{im}} \rightarrow \theta_{x_i|u_i} \tag{10}$$

$$\theta_{x_i|u_i} \rightarrow \lambda_{x_i} \tag{11}$$

$$\theta_{x_i|u_i} \rightarrow \lambda_{u_{i1}}, \cdots, \theta_{x_i|u_i} \rightarrow \lambda_{u_{im}} \tag{12}$$

Note that Definition 1 handles n-ary variables. Clauses (8) and (9) state that indicator variables are exclusive, while clauses (10)-(12) encode network's structure. Once we have encoded $\Pi G_{min}$, we compile $C_{min}$ into DNNF (denoted by $C_{DNNF}$) as shown in Figure 2. The resulting compiled base is qualified to be symbolic since it does not take into consideration any parameter value while encoding the network.

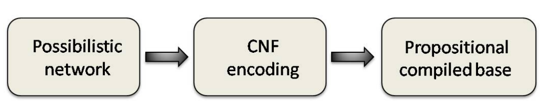



**Fig. 2.** Encoding and compilation steps.

**Example 2** *Let us re-consider the network of Figure 1. According to Algorithm 1.1, we should first encode* $\Pi G_{min}$ *as follows:* $C_{min} = (\lambda_{c_1} \vee \lambda_{c_2}) \wedge (\neg\lambda_{c_1} \vee \neg\lambda_{c_2}) \wedge (\lambda_{c_1} \wedge \lambda_{a_1} \wedge \lambda_{b_1} \rightarrow \theta_{c_1|a_1,b_1}) \wedge (\theta_{c_1|a_1,b_1} \rightarrow \lambda_{c_1}) \wedge (\theta_{c_1|a_1,b_1} \rightarrow \lambda_{a_1}) \wedge (\theta_{c_1|a_1,b_1} \rightarrow \lambda_{b_1}) \cdots$ *For lack of space, we have only written evidence clauses of the variable* $C$ *and* $\theta_{c_1|a_1,b_1}$*'s clauses. The full encoding contains 46 clauses.*

Computing efficiently the effect of observations in min-based possibilistic causal networks is ensured in the same spirit as the one given in [1]. While computing the effect of interventions as outlined Algorithm 1.1 requires a further step in which a simulation of mutilation is ensured under a compilation framework.

Let $do(x_I)$ be an intervention that forces the variable $X_I$ to take the value $x_I$, and $x$ an instantiation of some variables $X \subseteq V$, then computing $\Pi(x)$ given $do(x_I)$ is ensured by applying a fundamental function, i.e., *Computing* as follows:

1. Conditioning $C_{DNNF}$ on $x_I$ by setting $\theta_{x_i|u_i}$ of the variable of interest $X_I$ to $\top$ or $\bot$ depending on $x_i = x_I$ or $x_i \neq x_I$, resp.,
2. Conditioning $C_{DNNF}$ on $x$ by setting each $\lambda_{x_i}$ to:

$$\lambda_{x_i} = \begin{cases} \bot & \text{if } \exists \ x_j \in x \text{ s.t. } x_j \text{ and } x_i \text{ disagree on values (i.e., } x_i \nsim x) \\ \top & \text{otherwise (i.e., } x_i \sim x) \end{cases} \quad (13)$$

3. Decoding the mutilated and conditioned representation $C^m_{DNNF|x}$ to have a valued expression, denoted by $C^m_{\Pi-DNNF}$,
4. Computing $\Pi(x)$ by forgetting the remaining variables using the max operator (i.e., applying max-variable elimination).

The first step represents the mutilation phase under a compilation framework. Indeed, $C_{DNNF}$ is conditioned on $x_I$, as if we assign $\pi_m(x_I) = 1$ and $\forall x_i, x_i \neq x_I, \pi_m(x_i) = 0$. The motivational factor behind such operation resides in the symbolic compiled base that does not take into account parameters values. In fact, given interventions, new possibility degrees are generated which results in a new compiled base associated with the mutilated network as illustrated by Figure 3.

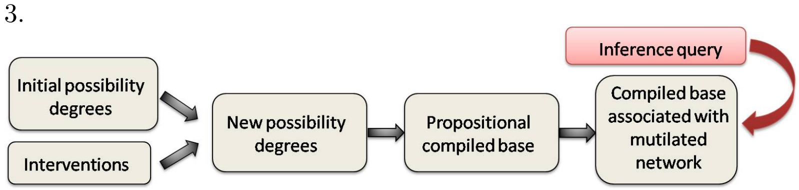



**Fig. 3.** The effect of interventions on compiled bases.

In the second step, we should condition $C_{DNNF}$ on $x$ using evidence indicators. The third step consists in decoding the mutilated representation by replacing $\vee$ and $\wedge$ by max and min, respectively and substituting each $\top$ and $\bot$ by their values. In fact, each $\top$ (resp. $\bot$) related to evidence indicators or $X_I$'s parameter variables is set to 1 (resp. 0), while each $\top$ corresponding to non $X_I$'s parameter variables is set to its possibility degree. Finally, we compute $\Pi(x)$

using $C^m_{\Pi-DNNF}$ by forgetting the remaining variables using the max operator. This operation, which is called *max-variable elimination* is the key for ensuring linear-time inference since $\Pi$-DNNF, that is a special case of VNNF, supports max-variable elimination [12].

It is worth pointing out that our approach takes advantage from the fact that the compiled base is restricted to a set of symbols without regard to numerical values as shown in Figure 2. In particular, for any new intervention, we can re-use the same original encoding with a simple updating of parameters values as depicted Figure 3. Hence, we can conclude that our method does not depend on interventions, i.e., even if we grow the number of interventions, the complexity is not altered since the simulation of mutilation, which is a conditioning operation is ensured in polynomial time and the computation of the effect of interventions is also polynomial with respect to the size of the compiled base.

Nevertheless, such encoding requires one variable per parameter, which is not the case if observations occur since parameters are stationary. Indeed, the so-called *local structure* enhancement related to equal parameters, used to reduce the set of added variables cannot be explored in mutilated $\Pi$-DNNF. More precisely, we cannot attribute the same propositional variable even for equal parameters within CPTs as in [1]. For instance, assuming that we have $\theta_{c_1|a_1,b_2} = \theta_{c_2|a_1,b_1} = 0.7$. After performing intervention $do(c_1)$, $\theta_{c_1|a_1,b_2}$ (resp. $\theta_{c_2|a_1,b_1}$) should be set to 1 (resp. 0) which is infeasible if we use the same $\theta$ for both of $\theta_{c_1|a_1,b_2}$ and $\theta_{c_2|a_1,b_1}$.

A minor enhancement can be performed after mutilation, which consists in merging $X_I$'s network parameters, i.e., each $\theta_{x_I|u_I}$, $\forall u_I$ can be replaced by $\theta_{x_I}$ since $X_I$ and $U_I$ are independent after mutilation.

**Algorithm 1.1.** Inference in Mutilated $\Pi$-DNNFs

Data: $\Pi G_{min}$, instance of interest $x$, evidence $e$, intervention $do(x_I)$

Result: $\Pi_c(x|e, do(x_I))$

**begin**

- Let $C_{min}$ be the CNF encoding obtained using equations (8)-(12)
- Let $C_{DNNF}$ be the compilation result of $C_{min}$
- $\Pi_c(x, e, do(x_I)) \leftarrow Computing(C_{DNNF}, (x, e, do(x_I)))$
- $\Pi_c(e, do(x_I)) \leftarrow Computing(C_{DNNF}, (e, do(x_I)))$
- **if** $\Pi_c(x, e, do(x_I)) \prec \Pi_c(e, do(x_I))$ **then** $\Pi_c(x|e, do(x_I)) \leftarrow \Pi_c(x, e, do(x_I))$
- **else** $\Pi_c(x|e, do(x_I)) \leftarrow 1$
- **return** $\Pi_c(x|e, do(x_I))$

**end**

**Proposition 2** *Let $\Pi G_{min}$ be a possibilistic network. Let $do(x_I)$ be an intervention that forces the variable $X_I$ to take the value $x_I$. Then, for any $x \in D_X$ and $e \in D_E$, we have $\Pi_c(x|e, do(x_I))$ (Algo. 1.1) $= \Pi_m(x|e, do(x_I))$ (Prop. 1).*

**Example 3** *Let us continue Example 3. Let $C$ be the variable in $\Pi G_{min}$ forced to take the value $c_1$ by the intervention $do(c_1)$. After encoding $\Pi G_{min}$, $C_{min}$ is then compiled into $C_{DNNF}$, from which we will compute for instance the effect of $do(c_1)$ and $a_1$ on $b_2$, namely computing $\Pi_c(b_2|a_1, do(c_1))$. We need to compute*

*both of $\Pi_c(b_2, a_1, do(c_1))$ and $\Pi_c(a_1, do(c_1))$. We start with $\Pi_c(b_2, a_1, do(c_1))$. First, we should set each $\theta_{c_1|u_i}$ (resp. $\theta_{c_2|u_i}$) to $\top$ (resp. $\bot$) $\forall u_i$ and condition $C_{DNNF}$ on $b_2$, $a_1$ and $do(c_1)$. The resulting $C^m_{DNNF|b_2,a_1,do(c_1)}$ is then decoded into $C^m_{\Pi-DNNF}$ as depicted Figure 4. For lack of space, we apply Max-VariableElimination to an excerpt of $C^m_{\Pi-DNNF} = min(\theta_{a_1}, \theta_{b_2}, \theta_{c_1})$.*
*Max-VariableElimination($C^m_{\Pi-DNNF}$, $\theta_{a_1}$) = max ($C^m_{\Pi-DNNF}|\theta_{a_1}$, $C^m_{\Pi-DNNF}|$ $\neg\theta_{a_1}$)= max(min(0.7, $\theta_{b_2}$, $\theta_{c_1}$),min(0, $\theta_{b_2}$, $\theta_{c_1}$)) = min(0.7, $\theta_{b_2}$, $\theta_{c_1}$),*
*$\Rightarrow$ Max-VariableElimination(min(0.7, $\theta_{b_2}$, $\theta_{c_1}$), $\theta_{b_2}$) = min(0.7, 0.4, $\theta_{c_1}$),*
*$\Rightarrow$ Max-VariableElimination(min(0.7, 0.4, $\theta_{c_1}$), $\theta_{c_1}$) = min(0.7, 0.4, 1) = 0.4.*
*This value corresponds exactly to the one computed in Example 1. $\Pi_c(a_1, do(c_1))$ is computed in the same spirit as $\Pi_c(b_2, a_1, do(c_1))$. Since $\Pi_c(b_2, a_1, do(c_1)) = 0.4$ $\prec \Pi_c(a_1, do(c_1)) = 0.7$, so $\Pi_c(b_2|a_1, do(c_1)) = 0.4$.*

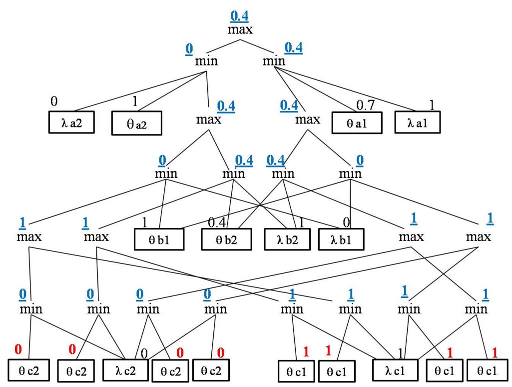



**Fig. 4.** Example of $C^m_{\Pi-DNNF}$.

## 4 Causal Inference using Compiled Possibilistic Bases

In [1], we proposed a purely possibilistic inference method which is not based on encoding probabilistic works grounded on compilation. The idea consists in transforming the binary network into a possibilistic knowledge base [3], encoding it into a CNF encoding by incorporating a set of additional variables corresponding exactly to the different weights of the base [5]. After that, a compilation step is required from which the inference query should be ensured in polynomial time. In this section, we will focus on how to enrich this method to compute efficiently the effect of both observations and interventions.

The key difference between handling only observations and handling both observations and interventions resides in the set of additional variables while encoding the possibilistic knowledge base. In fact, in a causal reasoning, even if we have the same degree for a set of formulae within the base, we cannot use the same propositional variable as we did for observations. In other terms, one variable per degree is used if we handle both observations and interventions as shown in the following proposition:

**Proposition 3** *Let $\Pi G_{min}$ be a possibilistic causal network. Let $\Sigma_{min}$ be its possibilistic knowledge base expressed by: $\Sigma_{min} = \Sigma_{X_1} \cup \Sigma_{X_2} \cup \cdots \cup \Sigma_{X_N}$, $\forall X_i \in V$*

*s.t.* $\Sigma_{X_i} = \{(\neg x_i \vee \neg u_i, a_i) : a_i = 1 - \Pi(x_i|u_i) \neq 0\}$*, then its CNF encoding is expressed by:* $K_\Sigma = \{\alpha_i \vee A_i : (\alpha_i, a_i) \in \Sigma_{min}\}$ *where* $A_i$ *is a propositional variable associated to each degree* $a_i$ *in* $\Sigma_{min}$*.*

It is important to note that even if we transform the network into a logic-based representation, causal links are not lost. Thanks to the parameters that allow us to encode network's structure.

**Example 4** *Let us re-consider the* $\Pi G_{min}$ *of Figure 1. Its CNF encoding is expressed by:* $K_\Sigma = \{(c_2 \vee a_2 \vee b_1 \vee A_1), (c_1 \vee a_2 \vee b_2 \vee A_2), (b_1 \vee A_3), (c_2 \vee a_1 \vee b_2 \vee A_4)), (a_2 \vee A_5), (c_2 \vee a_1 \vee b_1 \vee A_6)\}$ *s.t.* $A_1$*,* $A_2$*,* $A_3$*,* $A_4$*,* $A_5$ *and* $A_6$ *are propositional variables associated to 0.9, 0.7, 0.6, 0.5, 0.3 and 0.2, respectively.*

Computing the effect of both observations and interventions requires a clausal deduction test and a conditioning transformation, hence the CNF encoding $K_\Sigma$ should be compiled only once into the most succinct target compilation language that supports such operations. Our method is qualified to be flexible since it permits to exploit efficiently all the existing propositional compilers [1]. Given an instance of interest $x$, an evidence $e$ and an intervention $do(x_I)$ performed on $X_I$, then computing $\Pi_c(x|e, do(x_I))$ using the compiled base $K_c$ requires two steps as outlined Algorithm 1.2:

(i) $K_c$ should be mutilated by assigning the degree 1 (resp. 0) to variables $A_i$ corresponding to $\neg x_I$ (resp. $x_I$). The objective is to make the connection between mutilating $\Pi G_{min}$ and mutilating $K_c$ as shown in Figure 3. After mutilation, variables $A_i$ encoding equal degrees in $K_c$ can be merged into the same variable $B_j$. The new set of variables is denoted by $B = \{B_1, \ldots, B_g\}$ where $g$ represents the number of variables after merging.

(ii) We should first test if $K_c \nvDash B_1 \vee \neg e \vee \neg x_I$. If this deduction is not satisfied, we condition $K_c$ on $\neg B_1$ and then test if $K_c$ entails $\neg x$. If this is the case, we compute $\Pi_c(x|e, do(x_I))$, else we move to the next $B_j$ and we re-itere the same treatment. In the worst case, this computation is performed $g - 1$ times since the last variable $B_g$ corresponds to the degree 0.

It is worth pointing out that before the mutilation step we can not attribute the same $A_i$ even for equal degrees in $\Sigma_{min}$. For instance, assuming that we have the following formulae $(c_1 \vee a_1 \vee b_2, A_1)$ and $(c_2 \vee a_1 \vee b_1, A_1)$ such that $A_1$ encodes the degree 0.8. After performing intervention $do(c_1)$, we should set the degree 0 (resp. 1) to the $A_i$ corresponding to $(c_2 \vee a_1 \vee b_1, 0.8)$ (resp. $(c_1 \vee a_1 \vee b_2, 0.8)$), which is infeasible if we use the same variable $A_1$. It is also crucial to note that our method does not depend on the number of interventions. Thanks to the symbolic compiled base that allows us to update parameters values linearly regardless of the number of interventions.

**Proposition 4** *Let* $\Pi G_{min}$ *be a possibilistic network. Let* $do(x_I)$ *be an intervention that forces the variable* $X_I$ *to take the value* $x_I$*. Then, for any* $x \in D_X$ *and* $e \in D_E$*, we have* $\Pi_c(x|e, do(x_I))$ *(Algo. 1.2)* $= \Pi_m(x|e, do(x_I))$ *(Prop. 1).*

**Example 5** *We continue with Example 5. Let* $C$ *be the variable forced to take the value* $c_1$ *by the intervention* $do(c_1)$*, let* $a_1$ *be an observed evidence, then what*

**Algorithm 1.2.** Inference in Mutilated Possibilistic Bases

Data: $\Pi G_{min}$, instance of interest $x$, evidence $e$, intervention $do(x_I)$
Result: $\Pi_c(x|e, do(x_I))$
**begin**
  Let $\Sigma_{min}$ be the possibilistic base of $\Pi G_{min}$ using Proposition
  $K_\Sigma \leftarrow$ encoding($\Sigma_{min}$, $A$, $n$) using Proposition   % $A$ : *the set of propositional variables, n: the number of* $A_i$ *in* $A$
  Let $K_c$ be the compilation result of $K_\Sigma$
  Let $A_k$ be the set of propositional variables $A_i$ of $\neg x_i$
  **foreach** $A_i \in A_k$ **do** Degree of $A_i \leftarrow 1$
  Let $A_c$ be the set of $A_i$ of the variable of interest $X_i$ except $A_k$
  **foreach** $A_i \in A_c$ **do** Degree of $A_i \leftarrow 0$
  Let $B = \{B_1, \ldots, B_g\}$ be the new set of variables after merging
  i $\leftarrow$ 1, StopCompute $\leftarrow$ false, $\Pi_c(x|e, do(x_I)) \leftarrow 1$
  **while** *(*$K_c \nvDash B_i \vee \neg e \vee \neg x_I$*)* ***and*** *(*$i \leq g-1$*)* ***and*** *(StopCompute=false)* **do**
    $K_c \leftarrow$ condition ($K_c$, $\neg B_i$) using equation (6)
    **if** $K_c \vDash \neg x$ **then**
      *StopCompute* $\leftarrow$ true
      Let *degree*$(i)$ be the weight associated to $B_i$
      $\Pi_c(x|e, do(x_I)) \leftarrow$ 1-*degree*$(i)$
    **else** $i \leftarrow i+1$
  **return** $\Pi_c(x|e, do(x_I))$
**end**

*is the effect of* $do(c_1)$ *and* $a_1$ *on* $b_2$*? First, we should compile* $K_\Sigma$ *into DNNF as follows:* $K_c = \{[((((c_2 \wedge A_2) \vee c_1) \wedge b_1) \vee ((b_2 \wedge A_3) \wedge (c_2 \vee (c_1 \wedge A_1)))) \wedge A_5 \wedge a_1] \vee [a_2 \wedge ((b_1 \wedge (c_2 \vee (c_1 \wedge A_4))) \vee (b_2 \wedge A_3 \wedge (c_2 \vee (c_1 \wedge A_6))))]\}$*. Then, we should update the degree of* $A_k$=$\{A_2\}$ *from 0.7 to 1 and set the degree 0 for each variable in* $A_c = \{A_1, A_4, A_6\}$*. Merging variables gives us the new set of variables* $B = \{B_1(1), B_2(0.6), B_3(0.3), B_4(0)\}$ *which substitutes the set* $A$ *in* $K_c$ *as follows:* $K_c = \{[((((c_2 \wedge B_1) \vee c_1) \wedge b_1) \vee ((b_2 \wedge B_2) \wedge (c_2 \vee (c_1 \wedge B_4)))) \wedge B_3 \wedge a_1] \vee [a_2 \wedge ((b_1 \wedge (c_2 \vee (c_1 \wedge B_4))) \vee (b_2 \wedge B_2 \wedge (c_2 \vee (c_1 \wedge B_4))))]\}$*. We are now ready to compute* $\Pi_c(b_2|a_1, do(c_1))$*. The computation process requires two iterations which means that* $\Pi_c(b_2|a_1, do(c_1)) = 1 - degree(2) = 1 - 0.6 = 0.4$ *where* $degree(2)$ *designates the weight associated to* $B_2$*. This value corresponds exactly to the ones computed in Example 1 and Example 4.*

At this stage, a comparison study between the two proposed methods (i.e., mutilated $\Pi$-DNNFs and mutilated possibilistic bases) is crucial. It is clear that intuitively if we restrict our attention to the binary case, mutilated possibilistic bases are more compact than mutilated $\Pi$-DNNFs even if local structure is not exploited in both of them. In fact, mutilated possibilistic bases offer less variables and clauses since no formulae or clauses are associated to degrees equal to 1. Moreover, only one clause is encoded for each variable $A_i$ which is not the case for each parameter $\theta_{x_i|u_i}$ in mutilated $\Pi$-DNNFs. This note deserves to be confirmed by an experimental study.

## 5 Conclusion

This paper proposed compilation-based inference algorithms for min-based possibilistic causal networks. First, we proposed a compilation-based approach from which we compute efficiently the effect of both observations and interventions using mutilated $\Pi$-DNNFs. Then, we developed a possibilistic inference method dealing with mutilated possibilistic bases. Our methods are qualified as flexible since we compute the effect of observations and interventions from an already compiled base without a re-compilation cost. However, the inherent cost of interventions is expensive since we introduce different variables even for equal degrees, which is not the case for observations. This is the price to be paid if interventions occur. We have also noticed that intuitively, if we focus on the binary case, mutilated possibilistic bases are more compact than mutilated $\Pi$-DNNFs. Our future work consists in studying the representational point of view of interventions under a compilation framework, then comparing mutilated-based approaches with both augmented-based approaches and the well known junction tree.

## References


1. R. Ayachi, N. Ben Amor, S. Benferhat, and R. Haenni. Compiling possibilistic networks : Alternative approaches to possibilistic inference. In *Proceedings of 26th Conference on UAI*, pages 40–47. AUAI Press, 2010.
2. N. Ben Amor, S. Benferhat, and K. Mellouli. Anytime propagation algorithm for min-based possibilistic graphs. *Soft Comput.*, 8(2):150–161, 2003.
3. S. Benferhat, D. Dubois, L. Garcia, and H. Prade. On the transformation between possibilistic logic bases and possibilistic causal networks. *IJAR*, 29(2):135–173, 2002.
4. S. Benferhat and S. Smaoui. Possibilistic causal networks for handling interventions: A new propagation algorithm. In *AAAI*, pages 373–378, 2007.
5. S. Benferhat, S. Yahi, and H. Drias. On the compilation of stratified belief bases under linear and possibilistic logic policies. In *IJAR*, pages 2425–2430, 2007.
6. M. Cadoli and F. Donini. A survey on knowledge compilation. *AI Communications*, 10(3-4):137–150, 1997.
7. A. Darwiche. Decomposable negation normal form. *Journal of the ACM*, 48(4):608–647, 2001.
8. A. Darwiche. A logical approach to factoring belief networks. In *Proceedings of KR*, pages 409–420, 2002.
9. A. Darwiche. *Modeling and Reasoning with Bayesian Networks.* Cambridge University Press, New York, NY, USA, 1st edition, 2009.
10. A. Darwiche and P. Marquis. A knowledge compilation map. *Journal of Artificial Intelligence Research*, 17:229–264, 2002.
11. D. Dubois and H. Prade. Possibility theory. In Robert A. Meyers, editor, *Encyclopedia of Complexity and Systems Science*, pages 6927–6939. Springer, 2009.
12. H. Fargier and P. Marquis. On valued negation normal form formulas. In *Proceedings of the 20th IJCAI*, pages 360–365, San Francisco, CA, USA, 2007.
13. M. Goldszmidt and J. Pearl. Rank-based systems: A simple approach to belief revision, belief update, and reasoning about evidence and actions. In *KR*, pages 661–672, 1992.
14. J. Lang. Possibilistic logic: complexity and algorithms. In *Handbook on Defeasible Reasoning and Uncertainty Management Systems*, volume 5, pages 179–220. 2001.
15. J. Pearl. *Causality:models, reasoning, and inference.* Cambridge Univ. Press, 2000.