Fast Arc-Reversal

Cory J. Butz CORY.BUTZ@UREGINA.CA
Department of Computer Science, University of Regina, Canada

Anders L. Madsen ANDERS@HUGIN.COM
HUGIN EXPERT A/S, Aalborg, Denmark
Department of Computer Science, Aalborg University, Denmark

Jhonatan S. Oliveira JHONATAN@TALAWA.AI
Talawa.ai, San Jose, USA

Editors: J.H.P. Kwisthout & S. Renooij

Abstract

Fast arc-reversal (FAR) is proposed as a new exact inference algorithm in discrete *Bayesian networks* (BNs), merging favourable features of *Arc-reversal* (AR) and *Variable elimination* (VE). AR constantly maintains a sub-BN structure when rendering a variable barren via arc reversals, requiring more computational effort than VE, which sacrifices a sub-BN structure by directly eliminating a variable. We formally establish that FAR can *recover* a unique and sound sub-BN structure after consecutive variable eliminations. Experimental results on real-world benchmark networks empirically show a substantial improvement in the average run-time and variance of FAR compared to AR. We also suggest a novel method, called *d-contraction*, for graphically understanding FAR since FAR is not always the same as a sequence of arc reversals.

Keywords: Bayesian networks; exact inference; arc-reversal; variable elimination.

1. Introduction

Bayesian networks (BNs) (Pearl, 1988; Darwiche, 2009; Koller and Friedman, 2009) are a rich semantic modelling tool for managing uncertainty in complex domains. A BN consists of a *directed acyclic graph* (DAG) and a corresponding set of *conditional probability tables* (CPTs). The *probabilistic conditional independence* relations encoded in the DAG indicate that the product of the CPTs is a joint probability distribution. Probabilistic inference, also called belief propagation or belief update, answers a query, denoted $P(X|Y = y)$, by computing posterior probabilities of a target set of variables X through revising prior probabilities based on the observed values y of another set of variables Y, called evidence. It is important to continue research into improving BN inference algorithms since both exact and approximate inference are NP-hard tasks (Cooper, 1990; Dagum and Luby, 1993).

Eliminating the non-query variables from a BN is a fundamental task in inference. In this paper, we focus on two standard approaches, *Variable elimination* (VE) (Zhang and Poole, 1994; Dechter, 1996) and *Arc-reversal* (AR) (Olmsted, 1983; Shachter, 1986). VE eliminates a variable X_i by multiplying together all probability tables involving X_i and then summing X_i out of the product. AR eliminates a variable by transforming it into a leaf by reversing the arcs linking it to all its children in the BN. VE often involves less computation by directly eliminating a variable but sacrifices the clarity and richness of a BN sub-structure. Contrarily, AR stands out as an elegant inference algorithm due to its ability to preserve a sub-BN structure, albeit with the trade-off of increased computational overhead. Consequently, an opportunity exists to develop a novel hybrid inference algorithm that combines advantageous aspects of both VE and AR.

In this paper, we suggest *Fast arc-reversal* (FAR) as a new approach for exact inference in discrete BNs. It is formally shown that a unique and sound sub-BN structure can be recovered after a series of consecutive variable eliminations. The chain rule of probability, a fixed topological order, and probabilistic inference semantics are utilized for this purpose. Before eliminating a variable, AR and FAR must examine the same set of directed paths between the children of said variable, although they handle these paths differently. Empirical analysis of experimental results conducted on 25 real-world benchmark networks demonstrates that the FAR algorithm generally outperforms the AR algorithm in terms of computational speed and variance reduction on speed when eliminating variable sets for posterior probability computation. Lastly, we propose a graphical method, called *d-contraction*, to graphically understand FAR and compare and contrast it with AR's sequence of arc reversals.

The remainder of this paper is organized as follows. Background knowledge is reviewed in Section 2. Section 3 introduces FAR by showing how a BN sub-structure can be recovered. The theoretical foundation of FAR is established in Section 4. Section 5 describes experiments demonstrating the effectiveness of FAR. The graphical method d-contraction is put forth in Section 6. Section 7 contains conclusions drawn and outlines future work.

2. Background Knowledge

A Bayesian network $\mathcal{N} = (\mathcal{G}, \mathcal{P})$ has two components $\mathcal{G}$ and $\mathcal{P}$, where $\mathcal{G} = (V, E)$ is a DAG with vertices V and edges E, $\mathcal{P} = \{P(X_i \mid \text{pa}(X_i)) : X_i \in \mathcal{X}\}$ is a set of CPTs, $\mathcal{X}$ is a finite set of discrete variables with a one-to-one relationship between V and $\mathcal{X}$, and $\text{pa}(X_i)$ denotes the parents of X_i in $\mathcal{G}$. Subsets of $\mathcal{X}$ are denoted by uppercase letters X, Y, Z. We define (X_i, X_j) as the directed edge (arc) from X_i to X_j in $\mathcal{G}$. We let $\text{ch}(X_i)$, $\text{an}(X_i)$, and $\text{de}(X_i)$ respectively denote the children, ancestors, and descendants of one variable X_i in $\mathcal{G}$. These notions are naturally extended to sets of variables. A BN represents a factorization of a joint probability distribution $P(\mathcal{X})$ over $\mathcal{X}$ such that

$$P(\mathcal{X}) = \prod_{X_i \in \mathcal{X}} P(X_i \mid \text{pa}(X_i)).$$

One example BN is in Figure 1, where each variable X_i is depicted in figures by its subscript i for space considerations, and corresponding CPTs will not be shown.

We define a probability *potential* $\phi(X)$ as a non-negative and not-all-zero function over a set of variables X and a probability distribution $P(X)$ as a probability potential that sums to one (Shafer, 1996). The *domain* of probability potential $\phi(X)$, denoted $\text{dom}(\phi)$, is X. A CPT $\phi(X \mid Y)$ is called *singleton*, if $|X| = 1$, and *non-singleton*, if $|X| > 1$.

Inference is defined as computing $P(X \mid Y = y)$ posed to BN $\mathcal{N}$, where $Y = y$ is a set of variable instantiations and X is a set of target variables (a subset of non-evidence variables in $\mathcal{X}$). A variable X_i is *barren*, if $X_i \notin X$, X_i is not an evidence variable, and X_i only has barren descendants in $\mathcal{G}$, if any (Shachter, 1986).

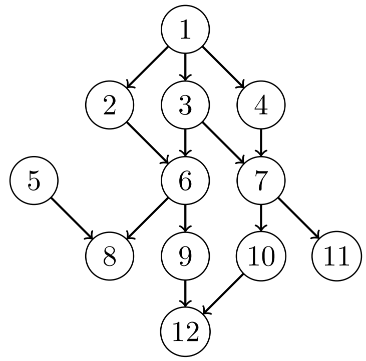


Figure 1: An example Bayesian network on $\mathcal{X} = \{X_1, X_2, \ldots, X_{12}\}$.

Arc-reversal (AR) (Olmsted, 1983; Shachter, 1986) removes a variable from $\mathcal{G}$ by reversing the arcs between the variable and its children and then building the CPTs corresponding to the modified DAG. Let (X_1, X_2) be an arc from X_1 to X_2 such that no other directed path from X_1 to X_2 exists in $\mathcal{G}$. The process of reversing the arc (X_1, X_2) amounts to performing the following three calculations:

$$\begin{aligned} P(X_1, X_2 \mid \mathrm{pa}(X_1) \cup \mathrm{pa}(X_2) \setminus \{X_1\}) &= P(X_1 \mid \mathrm{pa}(X_1)) P(X_2 \mid \mathrm{pa}(X_2)), \\ P(X_2 \mid \mathrm{pa}(X_1) \cup \mathrm{pa}(X_2) \setminus \{X_1\}) &= \sum_{X_1} P(X_1, X_2 \mid \mathrm{pa}(X_1) \cup \mathrm{pa}(X_2) \setminus \{X_1\}), \end{aligned} \tag{1}$$

$$P(X_1 \mid \mathrm{pa}(X_1) \cup \mathrm{pa}(X_2) \setminus \{X_1\} \cup \{X_2\}) = \frac{P(X_1, X_2 \mid \mathrm{pa}(X_1) \cup \mathrm{pa}(X_2) \setminus \{X_1\})}{P(X_2 \mid \mathrm{pa}(X_1) \cup \mathrm{pa}(X_2) \setminus \{X_1\})}. \tag{2}$$

Algorithm 1 specifies pseudo-code for how AR eliminates one variable X_i from a set of probability potentials Φ. To avoid introducing directed cycles during successive variable eliminations, edges to be added will always be oriented with respect to one fixed topological order $\prec$ of the given BN $\mathcal{N}$. In this paper, the topological order will always be fixed such that $X_i \prec X_j$, if $i < j$.

Example 1 *Consider any query where AR eliminates variable X_6 from the BN in Figure 1. To make X_6 a leaf, we can first reverse arc (X_6, X_8) as follows:*

$$\begin{aligned} P(X_6, X_8 | X_2, X_3, X_5) &= P(X_6 | X_2, X_3) P(X_8 | X_5, X_6), \\ P(X_8 | X_2, X_3, X_5) &= \sum_{X_6} P(X_6, X_8 | X_2, X_3, X_5), \end{aligned} \tag{3}$$

$$P(X_6 | X_2, X_3, X_5, X_8) = P(X_6, X_8 | X_2, X_3, X_5) / P(X_8 | X_2, X_3, X_5), \tag{4}$$

and then reverse arc (X_6, X_9):

$$\begin{aligned} P(X_6, X_9 | X_2, X_3, X_5, X_8) &= P(X_6 | X_2, X_3, X_5, X_8) P(X_9 | X_6), \\ P(X_9 | X_2, X_3, X_5, X_8) &= \sum_{X_6} P(X_6, X_9 | X_2, X_3, X_5, X_8). \end{aligned} \tag{5}$$

Algorithm 1: EliminateAR

Input: a variable X_i, a set of probability potentials Φ, a topological order $\prec$

Output: the updated set of probability potentials $\Phi^* \cup \Phi^*_{X_i}$

$\Phi_{X_i} \leftarrow \{\phi \in \Phi \mid X_i \in \text{dom}(\phi)\}$

$\Phi^* \leftarrow \Phi \setminus \Phi_{X_i}$

foreach $X_j \in \text{ch}(X_i)$ following $\prec$ **do**

  Compute $P^*(X_j \mid \text{pa}(X_j))$ in Equation (1)

  Compute $P^*(X_i \mid \text{pa}(X_i))$ in Equation (2)

  $\Phi^*_{X_i} \leftarrow \Phi^*_{X_i} \setminus \{P(X_i \mid \text{pa}(X_i)), P(X_j \mid \text{pa}(X_j))\} \cup \{P^*(X_i \mid \text{pa}(X_i)), P^*(X_j \mid \text{pa}(X_j))\}$

end

$\Phi^*_{X_i} \leftarrow \Phi^*_{X_i} \setminus \{P^*(X_i \mid \text{pa}(X_i))\}$

return $\Phi^* \cup \Phi^*_{X_i}$

Figure 2 depicts how AR maintains structure during the elimination of variable X_6 in Example 1. Note that when reversing the last arc (X_6, X_9), there is no need to build the new CPT for X_6, since X_6 will be immediately removed as barren in Figure 2(c).

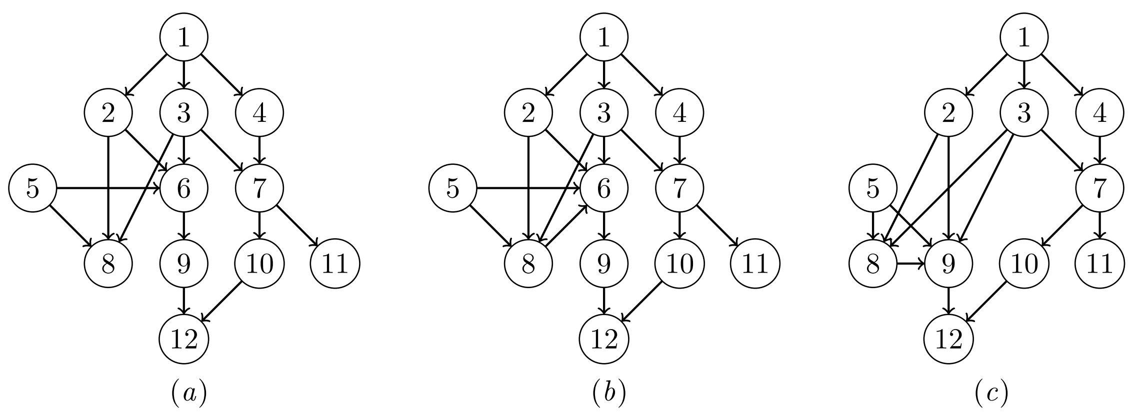


Figure 2: AR's Equations (3), (4) and (5) are shown in $(a), (b)$ and (c), respectively.

Variable elimination (VE) (Zhang and Poole, 1994; Dechter, 1996) is a method for eliminating a variable X_i from a set of probability potentials. It multiplies together all potentials with X_i in the domain and then sums out X_i from the product. Algorithm 2 eliminates a single variable X_i from a set Φ of potentials and returns the resulting set of potentials. Every multiplication and summation taken by VE yields a CPT, say of X given Y, although its probabilities may agree or disagree with the joint distribution $P(\mathcal{X})$, respectively called P-semantics, denoted $P(X|Y)$, and ϕ-semantics, denoted $\phi(X|Y)$ (Koller and Friedman, 2009; Butz and Yan, 2010).

Example 2 *VE eliminates variable X_6 from the BN in Figure 1 in a direct fashion:*

$$P(X_8, X_9 | X_2, X_3, X_5) \; = \; \sum_{X_6} P(X_6|X_2, X_3) P(X_8|X_5, X_6) P(X_9|X_6).$$

Algorithm 2: EliminateVE

Input: a variable X_i, a set of probability potentials Φ
Output: the updated set of probability potentials $\Phi \setminus \Phi_{X_i} \cup \{\phi_{X_i}\}$
$\Phi_{X_i} \leftarrow \{\phi \in \Phi \mid X_i \in \text{dom}(\phi)\}$
$\phi_{X_i} \leftarrow \sum_{X_i} \prod_{\phi \in \Phi_{X_i}} \phi$
return $\Phi \setminus \Phi_{X_i} \cup \{\phi_{X_i}\}$

3. Fast Arc-Reversal

We introduce *Fast arc-reversal* (FAR) as a new hybrid inference algorithm for exact inference in discrete BNs.

Algorithm 3 specifies pseudo-code for how FAR eliminates one variable X_i from a set of probability potentials Φ. VE is used first to eliminate X_i. Next, if VE created a non-singleton CPT, it is factorized into singleton CPTs with Algorithm 4, called ChainRule, which follows a fixed topological order $\prec$ to avoid introducing directed cycles.

Algorithm 3: Fast arc-reversal

Input: a variable X_i, a set of probability potentials Φ, a topological order $\prec$
Output: the updated set of probability potentials $\Phi^* \cup \Phi^*_{X_i}$
$\Phi_{X_i} \leftarrow \{\phi \in \Phi \mid X_i \in \text{dom}(\phi)\}$
$\Phi^* \leftarrow \Phi \setminus \Phi_{X_i}$
$\phi \leftarrow \sum_{X_i} \prod_{\phi \in \Phi_{X_i}} \phi$
$\Phi^*_{X_i} \leftarrow \phi$
if ϕ *is a non-singleton CPT* **then**
    $\Phi^*_{X_i} \leftarrow \textsc{ChainRule}(\phi, \prec)$
end
return $\Phi^* \cup \Phi^*_{X_i}$

For pedagogical purposes, structure recovery should be considered after each variable elimination. That is, variable elimination and structure recovery should be used alternatingly.

Example 3 *Let us recursively call FAR three times with $Z = \{X_6\}$, $Z = \{X_7\}$, and $Z = \{X_3\}$, to eliminate $\{X_6, X_7, X_3\}$, starting from the BN in Figure 1. To eliminate X_6, FAR computes the non-singleton CPT $P(X_8, X_9|X_2, X_3, X_5)$ by eliminating X_6 using VE as discussed in Example 2. Second, Algorithm 4 is called with non-singleton CPT $P(X_8, X_9|X_2, X_3, X_5)$ and $\prec$. Since $X_8 \prec X_9$, ChainRule computes*

$$P(X_8|X_2, X_3, X_5) = \sum_{X_9} P(X_8, X_9|X_2, X_3, X_5), \tag{6}$$

$$P(X_9|X_2, X_3, X_5, X_8) = P(X_8, X_9|X_2, X_3, X_5)/P(X_8|X_2, X_3, X_5). \tag{7}$$

Algorithm 4: ChainRule

Input: a non-singleton CPT $P(X_1, X_2, \ldots, X_k \mid Y)$, a topological order $\prec$

Output: singleton CPTs $\{P(X_1 \mid Y), P(X_2 \mid X_1, Y), \ldots, P(X_k \mid X_1, \ldots, X_{k-1}, Y)\}$

Let $X_1 \prec X_2 \prec \cdots \prec X_k$

for $i \leftarrow k$ **to** 2 **do**

  $P(X_1, \ldots, X_{i-1}|Y) \leftarrow \sum_{X_i} P(X_1, \ldots, X_i|Y)$

  $P(X_i|X_1, \ldots, X_{i-1}, Y) \leftarrow P(X_1, \ldots, X_i|Y)/P(X_1, \ldots, X_{i-1}|Y)$

end

return $\{P(X_1 \mid Y), P(X_2 \mid X_1, Y), \ldots, P(X_k \mid X_1, \ldots, X_{k-1}, Y)\}$

Figure 3(a) depicts the sub-BN defined by these CPTs. The subsequent call to FAR with $Z = \{X_7\}$ to eliminate variable X_7 with VE yields the non-singleton CPT:

$$P(X_{10}, X_{11}|X_3, X_4) \;=\; \sum_{X_7} P(X_7|X_3, X_4)P(X_{10}|X_7)P(X_{11}|X_7).$$

FAR calls ChainRule to equivalently factorize $P(X_{10}, X_{11}|X_3, X_4)$. Since $X_{10} \prec X_{11}$,

$$P(X_{10}|X_3, X_4) = \sum_{X_{11}} P(X_{10}, X_{11}|X_3, X_4),$$
$$P(X_{11}|X_3, X_4, X_{10}) = P(X_{10}, X_{11}|X_3, X_4)/P(X_{10}|X_3, X_4).$$

The resulting sub-BN is depicted in Figure 3(b). Calling FAR to eliminate $Z = \{X_3\}$ from Figure 3(b) yields the non-singleton CPT $P(X_8, X_9, X_{10}, X_{11}|X_1, X_2, X_4, X_5)$ by:

$$\sum_{X_3} P(X_3|X_1)P(X_8|X_2, X_3, X_5)P(X_9|X_2, X_3, X_5, X_8)P(X_{10}|X_3, X_4)P(X_{11}|X_3, X_4, X_{10}),$$

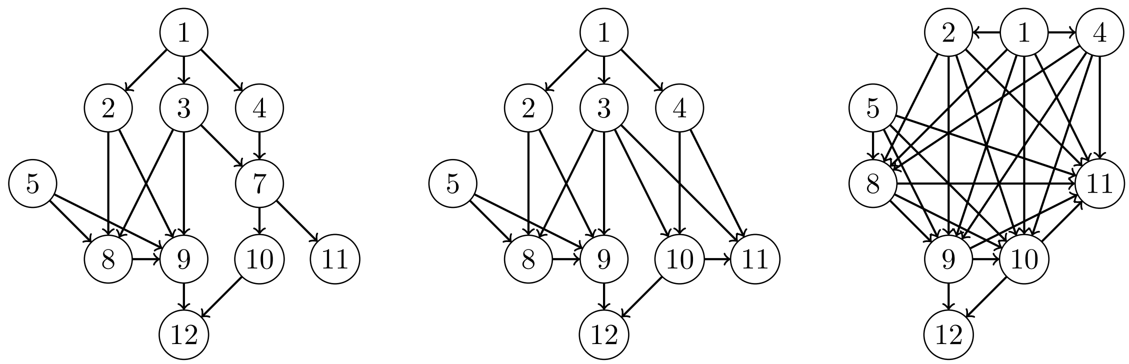
which ChainRule factories with $\prec$ as: $P(X_8|X_1, X_2, X_4, X_5)$, $P(X_9|X_1, X_2, X_4, X_5, X_8)$, $P(X_{10}|X_1, X_2, X_4, X_5, X_8, X_9)$, $P(X_{11}|X_1, X_2, X_4, X_5, X_8, X_9, X_{10})$. This yields the sub-BN depicted in Figure 3(c).

Example 3 shows that FAR can build a sub-BN structure by alternating between variable elimination and structure recovery. In fact, the reader can confirm that Figure 3(*a*) is the same sub-BN built by AR in Example 1. FAR's computation in Equation (6) and Equation (7) corresponds to AR's computation in Equation (3) and Equation (5), respectively.

More importantly, and more generally, FAR can recover a sub-BN structure after using VE for consecutive variable eliminations.

Example 4 *Consider how FAR eliminates $Z = \{X_3, X_6, X_7\}$ from the BN in Figure 1, where the elimination order is $\sigma = (X_6, X_7, X_3)$. The first phase of FAR involves the elimination of these variables using VE, yielding the following non-singleton CPTs:*

$$P(X_8, X_9|X_2, X_3, X_5) \;=\; \sum_{X_6} P(X_6|X_2, X_3)P(X_8|X_5, X_6)P(X_9|X_6),$$
$$P(X_{10}, X_{11}|X_3, X_4) \;=\; \sum_{X_7} P(X_7|X_3, X_4)P(X_{10}|X_7)P(X_{11}|X_7),$$



(a) After X_6 (b) After X_7 (c) After X_3

Figure 3: Alternating variable elimination and structure recovery.

and $P(X_8, X_9, X_{10}, X_{11}|X_1, X_2, X_4, X_5)$ *by*

$$\sum_{X_3} P(X_3|X_1)P(X_8, X_9|X_2, X_3, X_5)P(X_{10}, X_{11}|X_3, X_4).$$

Now that VE has eliminated all variables in Z*, the second phase of FAR begins. ChainRule is called with the non-singleton CPT* $P(X_8, X_9, X_{10}, X_{11}|X_1, X_2, X_4, X_5)$ *and* $\prec$*, yielding the following four singleton CPTs:* $P(X_8|X_1, X_2, X_4, X_5)$*,* $P(X_9|X_1, X_2, X_4, X_5, X_8)$*,* $P(X_{10}|X_1, X_2, X_4, X_5, X_8, X_9)$*, and* $P(X_{11}|X_1, X_2, X_4, X_5, X_8, X_9, X_{10})$*.*

The reader can verify that AR's elimination of X_6, X_7, X_3 from the BN in Figure 1 yields the sub-BN in Figure 3(c). The key point of Example 4 is that FAR built the same sub-BN, but by applying VE three times consecutively, and then utilizing ChainRule.

4. Theoretical Foundation

In this section, we establish the theoretical foundation of FAR, including its soundness and uniqueness, after discussing similarities and differences between AR and FAR.

The next example shows that sometimes FAR and AR can differ.

Example 5 *FAR's and AR's elimination of* X_2 *from the BN* $\mathcal{N}$ *in Figure 4(*a*) are depicted in Figure 4(*b*) and Figure 4(*c*), respectively.*

Example 5 illustrates that while FAR may recover AR's sub-BN structure, there is no guarantee that it will always do so.

Both AR and FAR must detect all directed paths between the children of the variable being eliminated, which we call "second paths." These paths can be partitioned as a dichotomy and have different lasting effects on probabilistic inference semantics (Butz and Yan, 2010). Let us first consider directed paths between children of the variable being eliminated involving only children of said variable. If such a path exists, AR must reverse the arcs between children in a certain order so as to not create a directed cycle and a ϕ-CPT.

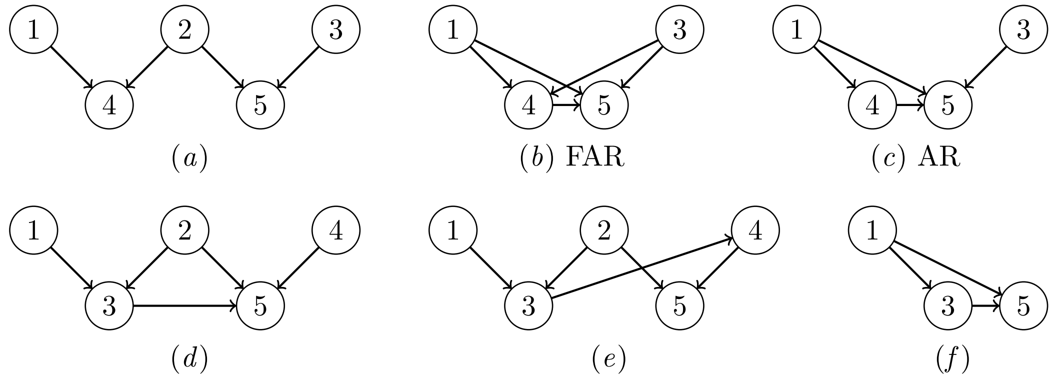


Figure 4: When eliminating X_2 from the BN in (a), FAR's sub-DAG in (b) can be different from AR's sub-DAG in (c). AR restricts the arc reversal order when eliminating X_2 in (d) and (e). FAR eliminates X_2 in (d) without restriction but cannot eliminate only $\{X_2\}$ in (e). FAR and AR both build (f) from (e) when using P-elimination order $\sigma = (X_4, X_2)$.

Example 6 *Consider the elimination of X_2 from the BN $\mathcal{N}$ in Figure 4(d), recognizing the directed path (X_3, X_5) between children of X_2. AR imposes the restriction that arc (X_2, X_3) be reversed first; otherwise, AR would generate a ϕ-CPT with its first multiplication:*

$$\phi(X_2, X_5 \mid X_3, X_4) \;=\; P(X_2)P(X_5 \mid X_2, X_3, X_4).$$

FAR can take this product since P-semantics are recovered with the next multiplication:

$$P(X_2, X_3, X_5 \mid X_1, X_4) \;=\; P(X_3 \mid X_1, X_2)\phi(X_2, X_5 \mid X_3, X_4).$$

Example 6 shows that when the "second path" only involves children of the variable being eliminated, then AR takes care to reverse arcs in a certain order, while FAR can eliminate the variable without special consideration.

Now consider the second type of directed path between children of the variable X_i being eliminated. Here, the directed path involves a node that is not a child of X_i. If such a path exists, FAR's elimination of X_i in isolation yields a ϕ-CPT.

Example 7 *Consider the elimination of X_2 from the BN $\mathcal{N}$ in Figure 4(e), noting the directed path from child X_3 to child X_5 involving non-child X_4. As discussed in (Koller and Friedman, 2009; Butz and Yan, 2010), the elimination of X_2 in isolation by VE yields,*

$$\phi(X_3, X_5 \mid X_1, X_4) \;=\; \sum_{X_2} P(X_2)P(X_3 \mid X_1, X_2)P(X_5 \mid X_2, X_4).$$

As Example 7 shows that FAR may construct ϕ-CPTs, FAR is not guaranteed to build a sound sub-BN using arbitrary elimination orders. It is important to emphasize that multiplying together all remaining CPTs in the factorization always yields a correct marginal distribution within the FAR framework. This is because FAR employs VE for marginalization and equivalently replaces non-singleton CPTs with singleton CPTs. For example, in Example 7, a correct marginal is maintained after applying ChainRule on $\phi(X_3, X_5 \mid X_1, X_4)$:

$$P(X_1, X_3, X_4, X_5) \;=\; P(X_1)\phi(X_3 \mid X_1, X_4)P(X_4 \mid X_3)\phi(X_5 \mid X_1, X_3, X_4).$$

In order to guarantee that FAR constructs sound sub-BNs, the elimination order must be a P-elimination order. A *P-elimination order* (Butz et al., 2023) has the property that before eliminating a variable X_i, all non-children of X_i appearing on directed paths between children of X_i are eliminated first.

Example 8 *When eliminating $Z = \{X_2, X_4\}$ using P-elimination order $\sigma = (X_4, X_2)$ from the BN in Figure 4(e), FAR and AR build the same sound sub-BN in Figure 4(f).*

Henceforth, we assume that a P-elimination order is always used when FAR is to build a sound sub-BN. We now present our main result.

Theorem 1 *Let $\mathcal{N} = (\mathcal{G}, \mathcal{P})$ be a BN defining joint distribution $P(\mathcal{X})$, $\prec$ be a fixed topological order of $\mathcal{G}$, and $Z \subseteq \mathcal{X}$ be a subset of variables to be eliminated following P-elimination order σ. Then FAR constructs a unique, sound sub-BN for the marginal distribution $P(\mathcal{X} \setminus Z)$.*

Proof Let X_l be the first variable eliminated. Let X_l have k children, $X_1, \ldots, X_k$. By assumption, a directed path does not exist from any child of X_l to another child of X_l involving a variable that is not a child of X_l. By (Butz and Yan, 2010), VE's product of these $k+1$ CPTs is $P(X_1, \ldots, X_k, X_l \mid pa(X_1, \ldots, X_k, X_l) \setminus \{X_1, \ldots, X_k, X_l\})$. VE's marginalization of X_l yields $P(X_1, \ldots, X_k \mid pa(X_1, \ldots, X_k, X_l) \setminus \{X_1, \ldots, X_k, X_l\})$. By the chain rule and the fixed topological order $\prec$, this non-singleton CPT is equal to the product of the following k singleton CPTs:

$$\begin{aligned}
&P(X_1 \mid pa(X_1, \ldots, X_k, X_l) \setminus \{X_1, \ldots, X_k, X_l\}),\\
&P(X_2 \mid \{X_1\} \cup pa(X_1, \ldots, X_k, X_l) \setminus \{X_1, \ldots, X_k, X_l\}),\\
&\qquad\vdots\\
&P(X_k \mid \{X_1, \ldots, X_{k-1}\} \cup pa(X_1, \ldots, X_k, X_l) \setminus \{X_l\}).
\end{aligned}$$

Note that no conditional independence relations are introduced in the chain rule factorization. Now, every variable in $\mathcal{X} \setminus X_l$ has a singleton CPT, and the directed graph defined by these singleton CPTs is acyclic since all new edges are directed with respect to $\prec$. Thus, FAR constructs a sound sub-BN for the marginal distribution $P(\mathcal{X} \setminus X_l)$. A similar argument holds for the other variables in Z. Uniqueness immediately follows from $\prec$ and σ. ■

Theorem 1 states that FAR can always construct a unique and sound sub-BN.

By its construction, however, FAR added an extraneous arc (X_3, X_4) in Figure 4(b). This arc destroys the unconditional independence of X_3 and X_4, denoted $I(X_3, \emptyset, X_4)$, that holds in the original BN in Figure 4(a). AR did not add this arc in Figure 4(c). It is important to observe that VE sacrificed $I(X_3, \emptyset, X_4)$ when eliminating X_2:

$$P(X_4, X_5 \mid X_1, X_3) \;=\; \sum_{X_2} P(X_2)P(X_4 \mid X_1, X_2)P(X_5 \mid X_2, X_3).$$

FAR is unable to recover $I(X_3, \emptyset, X_4)$ when applying the chain rule

$$P(X_4, X_5 \mid X_1, X_3) \;=\; P(X_4 \mid X_1, X_3)P(X_5 \mid X_1, X_3, X_4).$$

Future work will investigate improving FAR by not adding superfluous arcs.

5. Experimental Analysis

In this section, we describe the experimental analysis performed to compare AR and FAR. The experiment involves 25 BNs of different complexity taken from the literature. The objective of the experimental analysis is to investigate and compare the performance impact of AR and FAR as the algorithm used for eliminating sets of variables during message passing in Simple Propagation (Butz et al., 2016).

In each case, an optimal triangulation in terms of total clique state-space size (TSS) has been generated for each network using the *total-weight* algorithm of the HUGIN Decision Engine (Madsen et al., 2005). Information on the 25 BNs and the corresponding junction trees can be found in Table 1 (columns two and three). Networks of different sizes and complexity are considered in the experimental analysis.

The empirical evaluation is performed on a desktop computer running Red Hat Enterprise Linux 7.9 with a six-core Intel (TM) i7-5820K 3.3GHz processor and 64 GB RAM. The computer has six physical cores and twelve logical cores. Computation time is measured as the elapsed (wall-clock) time in seconds and covers both message passing and computation of marginals. Note that the semantics of FAR's CPTs are not considered in the experiments as intermediate ϕ-CPTs do not disturb the computation of posterior probabilities.

Table 1 also shows the experimental results, where random evidence is propagated in each BN. For each network, 100 sets of randomly generated evidence are propagated. The same evidence is used for each algorithm. Each algorithm has a separate column, i.e., FAR and AR. The lowest average run-time for each BN is highlighted in bold.

Table 1 empirically shows that FAR had the unique lowest cost in 21 cases compared to AR, which had a unique lowest cost in 1 case. FAR and AR are tied in 3 networks.

6. d-Contraction

We introduce *d-contraction* to graphically understand FAR akin to visualizing AR as a sequence of arc reversals.

Definition 2 ***(d-contraction)***[1] *Let $\mathcal{N}$ be a BN on $\mathcal{X}$ and $\prec$ be the fixed topological order of variables in $\mathcal{N}$. Let $Z \subseteq \mathcal{X}$ be a subset of variables to be eliminated. Following a P-elimination order σ, graphically eliminate variable X_l by adding a directed edge (X_i, X_j) between the following pairs of variables, provided $X_i \prec X_j$: (i) from every current parent of X_l to every current child of X_l; (ii) from every current spouse (parent of a common child) of X_l to every current child of X_l; and (iii) between every pair of current children of X_l. Remove X_l and its incident edges from the current sub-DAG.*

1. d-contraction is intended as a Bayesian variant of contraction in graph theory in the spirit of d-separation and separation, and is not to be confused with the Contraction semi-graphoid inference axiom.

Network	$\mathcal{X}$	log(TSS)	$\mu(FAR)$	$\sigma^2(FAR)$	$\mu(AR)$	$\sigma^2(AR)$
3nt	58	3.45	**0.01**	0.00	0.02	0.00
ADAPT_1	133	1.98	**0.04**	0.00	0.05	0.03
Amirali_network	681	6.92	**0.38**	0.20	0.55	0.41
andes	223	4.82	**0.13**	0.06	0.19	0.11
Barley	48	6.86	**0.08**	0.13	0.12	0.20
cc145	145	3.01	**0.08**	0.04	0.13	0.09
cc245	245	5.42	**0.16**	0.08	0.27	0.16
Diabetes	413	4.93	**0.29**	0.26	1.05	1.59
food	109	6.48	**0.14**	0.17	0.16	0.17
hailfinder	56	3.51	**0.02**	0.00	0.03	0.00
Heizung.	44	7.62	**0.19**	0.45	0.28	0.63
Hepar_II	70	2.58	**0.02**	0.00	0.05	0.03
KK	50	6.76	**0.08**	0.10	0.09	0.10
medianus	56	5.73	**0.03**	0.03	0.04	0.04
Mildew	35	6.10	**0.04**	0.04	0.05	0.05
Munin1	189	7.58	**0.48**	1.12	1.41	4.23
oow_bas	27	5.71	**0.03**	0.03	**0.03**	0.03
oow_solo	40	6.22	**0.07**	0.09	0.08	0.10
oow	33	6.31	**0.07**	0.09	**0.07**	0.09
pathfinder	109	4.51	**0.11**	0.08	0.17	0.14
powerplant	46	1.91	**0.01**	0.00	**0.01**	0.00
ship	50	6.61	0.16	0.29	**0.15**	0.24
system_v57	85	4.84	**0.05**	0.03	0.07	0.04
Water	32	5.77	**0.05**	0.06	0.07	0.07
win95pts	76	2.71	**0.03**	0.00	0.05	0.03

Table 1: Average time cost in seconds propagating random evidence in 25 real-world Bayesian networks. Lowest costs are specified in bold

Example 9 *Consider applying d-contraction to eliminate* $Z = \{X_3, X_6, X_7\}$ *from the BN in Figure 1 following the P-elimination order* $\sigma = (X_6, X_7, X_3)$*. For* X_6*, in step (i), when considering parents* X_2 *and* X_3 *and children* X_8 *and* X_9*, add directed edges* (X_2, X_8)*,* (X_2, X_9)*,* (X_3, X_8)*, and* (X_3, X_9)*. In step (ii), add directed edge* (X_5, X_9) *from the spouse* X_5 *of* X_6 *to the child* X_9 *of* X_6*. In step (iii), add directed edge* (X_8, X_9) *between children* X_8 *and* X_9*, since* $X_8 \prec X_9$*. Finally, delete vertex* X_6 *and all incident edges* (X_2, X_6)*,* (X_3, X_6)*,* (X_6, X_8)*, and* (X_6, X_9)*. The resulting sub-DAG is depicted in Figure 3(a).*

The reader can verify that subsequently removing X_7 *with d-contraction gives the sub-BN in Figure 3(b). Next, to remove* X_3*, observe that* X_3*'s children have changed from the original BN. Thus, in step (i), when considering parent* X_1 *and children* $ch(X_3) = \{X_8, X_9, X_{10}, X_{11}\}$*, add directed edges* $(X_1, X_8), (X_1, X_9), (X_1, X_{10})$*, and* (X_1, X_{11})*. In step (ii), with* X_2*,* X_4*, and* X_5 *each being a spouse, add directed edges* (X_2, X_{10})*,* (X_2, X_{11})*,* (X_4, X_8)*,* (X_4, X_9)*,* (X_5, X_{10})*, and* (X_5, X_{11})*. In step (iii), add directed edges* (X_8, X_{10})*,* (X_8, X_{11})*,* (X_9, X_{10})*, and* (X_9, X_{11})*. Finally, delete vertex* X_3 *and all incident edges* (X_1, X_3)*,* (X_3, X_8)*,* (X_3, X_9)*,* (X_3, X_{10}) *and* (X_3, X_{11})*. The resulting sub-DAG is depicted in Figure 3(c).*

The next example highlights how d-contraction can differ graphically from a series of arc reversals with AR.

Example 10 *Let us graphically eliminate* X_2 *in Figure 4(a) using d-contraction. In step (ii), arc* (X_3, X_4) *is added from the spouse* X_3 *to the child* X_4 *of* X_2*. In step (iii), arc* (X_4, X_5) *is added from* X_4 *to the other child* X_5 *of* X_2*, yielding the sub-BN in Figure 4(b). As previously mentioned, the series of arc reversals done by AR to eliminate* X_2 *in Figure 4(a) gives Figure 4(c), which does not involve adding an arc from the spouse* X_3 *to the child* X_4*.*

Example 10 highlights that d-contraction and a series of arc reversals are two alternative ways of graphically visualizing sound sub-BN construction during variable elimination.

7. Conclusions

We introduced a novel BN exact inference algorithm, called *Fast arc-reversal* (FAR), that falls between AR and VE while incorporating favourable features of each. Whereas its predecessor, AR, constantly maintains a sub-BN structure as in Figure 2, FAR recovers a sub-BN structure after running VE, as in Figure 3(*a*), for instance. To ensure clarity in pedagogy, we presented in Example 3 how variable elimination and structure recovery can be performed in an alternating sequence. More importantly, our main result, Theorem 1, states that FAR can recover a unique and sound sub-BN after consecutive variable eliminations. In our experimental results on 25 benchmark real-world BNs, FAR was faster than AR in 21 cases, tied AR in 3 cases, and was slower than AR once, as shown in Table 1. As well, FAR typically exhibits lower variance. Lastly, we put forth *d-contraction* to graphically understand FAR, as FAR is not necessarily the same as a series of arc reversals. Future work will investigate preventing FAR from adding superfluous arcs as mentioned in Section 4.

References

C. J. Butz and W. Yan. The semantics of intermediate CPTs in variable elimination. In *Proceedings of the Fifth European Workshop on Probabilistic Graphical Models*, pages 41–49, 2010.

C. J. Butz, J. S. Oliveira, A. E. dos Santos, and A. L. Madsen. Bayesian network inference with Simple Propagation. In *Proceedings of the Twenty-Ninth International Florida Artificial Intelligence Research Society Conference*, pages 650–655, 2016.

C. J. Butz, J. S. Oliveira, A. dos Santos, A. Norris, and K. Meredith. Upward pass semantics in arithmetic circuit inference. In *Proceedings of the Thirty-Six Canadian Artificial Intelligence Conference*, 2023.

G. Cooper. The computational complexity of probabilistic inference using Bayesian belief networks. *Artificial Intelligence*, 42(2-3):393–405, 1990.

P. Dagum and M. Luby. Approximating probabilistic inference in Bayesian belief networks is np-hard. *Artificial Intelligence*, 60(1):141–153, 1993.

A. Darwiche. *Modeling and Reasoning with Bayesian Networks.* Cambridge University Press, Los Angeles, CA, 2009.

R. Dechter. Bucket elimination: A unifying framework for reasoning. In *Proceedings of the Twelfth Conference on Uncertainty in Artificial Intelligence*, pages 211–219, 1996.

D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques.* MIT Press, Cambridge, MA, 2009.

A. L. Madsen, F. Jensen, U. Kjærulff, and M. Lang. HUGIN - The Tool for Bayesian Networks and Influence Diagrams. *International Journal on Artificial Intelligence Tools 14*, 3:507–543, 2005.

S. Olmsted. *On Representing and Solving Decision Problems.* PhD thesis, Stanford University, 1983.

J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan Kaufmann, San Francisco, CA, 1988.

R. D. Shachter. Evaluating influence diagrams. *Operations Research*, 34(6):871–882, 1986.

G. Shafer. *Probabilistic Expert Systems.* Philadelphia: Society for Industrial and Applied Mathematics, Philadelphia, PA, 1996.

N. L. Zhang and D. Poole. A simple approach to Bayesian network computations. In *Proceedings of the Tenth Canadian Artificial Intelligence Conference*, pages 171–178, 1994.