Equivalences between Maximum a Posteriori Inference in Bayesian Networks and Maximum Expected Utility Computation in Influence Diagrams

Denis D. Mauá

Universidade de São Paulo,
São Paulo, Brazil
denis.maua@usp.br

Abstract. Two important tasks in probabilistic reasoning are the computation of the maximum posterior probability of a given subset of the variables in a Bayesian network (MAP), and the computation of the maximum expected utility of a strategy in an influence diagram (MEU). Despite their similarities, research on both problems have largely been conducted independently, with algorithmic solutions and insights designed for one problem not (trivially) transferable to the other one. In this work, we show constructively that these two problems are equivalent in the sense that any algorithm designed for one problem can be used to solve the other with small overhead. These equivalences extend the toolbox of either problem, and shall foster new insights into their solution.

Keywords: Bayesian networks, maximum a posteriori inference, influence diagrams, maximum expected utility.

1 Introduction

Maximum a posteriori inference (MAP) consists in finding a configuration of a certain subset of the variables that maximizes the posterior probability distribution induced by a Bayesian network [30]. MAP has applications, for example, in diagnostic systems and classification of relational and sequential data [15]. Solving MAP is computationally difficult, and the literature contains a plethora of approximate solutions, a few examples being the works in [4,29,17,18,13,21,26]

Influence diagrams extend Bayesian networks with preferences and actions to cope with decision making situations [10,12]. The maximum expected utility problem (MEU) is to select a mapping from observations to actions that maximizes the expected utility as defined by an influence diagram. MEU appears, for example, in troubleshooting and active sensing [12]. Although MEU is computationally difficult to solve, it counts with a large number of approximate solutions, for example, the works in [32,14,20,5,24,19,16,9,7,6].

L.C. van der Gaag and A.J. Feelders (Eds.): PGM 2014, LNAI 8754, pp. 318–333, 2014.

The MAP and MEU problems are closed for the complexity class NP^{PP} [30,5], which implies that any algorithm designed to solve one problem can *in principle* be used to solve the other.[1] Moreover, both problems are NP-complete when the treewidth of the underlying diagram is assumed bounded [3,22,23].[2] In practice, however, these two problems have been investigated independently, with a few similarities arising in the design of algorithms such as the use of clique-tree structures and message-passing for fast probabilistic inference [20,21].

In this work we provide *constructive* proofs of the equivalences between these two problems. We start by presenting background knowledge on graphs (Sec. 2), Bayesian networks (Sec. 3) and influence diagrams (Sec. 4), and formalizing the MAP and MEU problems. Then, we design a polynomial-time reduction that maps MAP problems into MEU problems (Sec. 5). We show that the reduction increases the treewidth by at most four, which makes the reduction closed in NP. We proceed to build a polynomial-time reduction of MEU into MAP problems (Sec. 6). The reduction increases treewidth by at most five, being also closed in NP. These reductions enlarge the algorithmic toolbox of either problem, and shall bring new insights into the design of new algorithms. We conclude with an overview of the results and a brief discussion on some shortcomings of the reductions developed here (Sec. 7).

2 Some Useful Concepts from Graph Theory

Consider a directed graph with nodes X and Y. A node X is a *parent* of a Y if there is an arc going from X to Y, in which case we say that Y is a *child* of X. The *in-degree* of a node is the number of its parents. We denote the parents of a node X by $\text{pa}(X)$ and its children by $\text{ch}(X)$. The family of a node comprises the node itself and its parents. A *polytree* is a directed acyclic graph (DAG) which contains no undirected cycles. A DAG is *loopy* if it is not a polytree. Polytrees are important, as they are among the simplest structures, and probabilistic inference can be performed efficiently in some polytree-shaped Bayesian networks.

The *moral graph* of a DAG is the undirected graph obtained by connecting nodes with a common child and dropping arc directions. The moral graph of a DAG might contain (undirected) cycles even when the DAG itself does not (e.g., any polytree with maximum in-degree greater than one).

A *tree decomposition* of an undirected graph G is a tree T such that

1. each node i associated to a subset $\mathcal{X}_i$ of nodes in G;
2. for every edge X-Y of G there is a node i of T whose associated node set $\mathcal{X}_i$ contains both X and Y;
3. for any node X in G the subgraph of T obtained by considering only nodes whose associate set contain X is a tree.

[1] We assume here that the number of incoming arcs into any decision node in an influence diagram is logarithmically bounded by the number of variables, which limits the size of strategies to a polynomial in the input size.

[2] The treewidth of a graph is a measure of its similarity to a tree.

The third property is known as the running intersection property. A *clique* is a set of pairwise connected nodes of an undirected graph. Any tree decomposition of a graph contains every clique of it included in some of the associated node sets [2]. The *treewidth* of a tree decomposition is the maximum cardinality of a node set $\mathcal{X}_i$ associated to a node of it minus one. The treewidth of a graph G is the minimum treewidth over all tree decompositions of it. The treewidth of a directed graph is the treewidth of its corresponding moral graph. Polytrees have treewidth given by the maximum in-degree of a node.

The elimination of a node X from a graph G produces a graph G' by removing X (and its incident arcs) and pairwise connecting all its neighbors. A node is *simplicial* if all its neighbors are pairwise connected. Eliminating a simplicial node is the same as simply removing it (and its incident arcs) from the graph. Let G be a graph of treewidth κ, and G' be the graph of treewidth κ' obtained from G by eliminating a node X of degree d. Then κ is at most $\max\{\kappa', d\}$, being exactly that when X is simplicial [2]. By removing arcs or nodes we generate graphs whose treewidth are not larger than the original graph.

3 Bayesian Networks and the MAP Problem

A Bayesian network consists of a DAG G over a set of variables $\mathbf{X}$ and a set of conditional probability assessments $P(X = x|\text{pa}(X) = \pi)$, one assessment for every variable X in $\mathbf{X}$ and configurations x and π of X and $\text{pa}(X)$, respectively. The DAG encodes a set of Markov conditions: every variable is independent of its non-descendant non-parents given its parents. These conditions induce a joint probability distribution over the variables that factorizes as $P(\mathbf{X}) = \prod_{X \in \mathbf{X}} P(X|\text{pa}(X))$.

The treewidth of a Bayesian network is defined as the treewidth of its underlying DAG. When using tree decompositions of Bayesian networks we refer to the sets associated to nodes of the tree as variable sets, since every node is identified with a variable. Probabilistic inference can be performed in time at most exponential in the treewidth of the network, hence in polynomial-time if treewidth is bounded [15].

Let $(\mathbf{M}, \mathbf{E}, \mathbf{H})$ denote a partition of $\mathbf{X}$ and $\hat{\mathbf{e}}$ be an assignment to $\mathbf{E}$. The set $\mathbf{M}$ contains *MAP variables*, whose values we would like to infer; the set $\mathbf{E}$ contains *evidence variables*, whose values are known to be (i.e., they are fixed at) $\hat{\mathbf{e}}$; at last, the set $\mathbf{H}$ contains *hidden* variables, whose values we ignore (i.e., they are marginalized out). The MAP problem consists in computing the *value*

$$\max_{\mathbf{m}} P(\mathbf{M}=\mathbf{m}, \mathbf{E}=\hat{\mathbf{e}}) = \max_{\mathbf{m}} \sum_{\mathbf{H}} P(\mathbf{M}=\mathbf{m}, \mathbf{E}=\hat{\mathbf{e}}, \mathbf{H}) \,. \tag{1}$$

A configuration $\mathbf{m}^*$ which maximizes the equation above is known as a maximum a posteriori configuration or posterior mode, as it also maximizes the posterior probability distribution $P(\mathbf{M}|\mathbf{E}=\hat{\mathbf{e}})$. We can compute $\mathbf{m}^*$ by recursively solving MAP problems as follows. First, solve the MAP problem (call this problem unconstrained). Label all MAP variables free and repeat the following procedure until no free variables remain: Select a free variable M_i and clamp it at a value m_i^* such that the MAP problem with $M_i = m_i^*$ returns the same value as the unconstrained problem; label this variable fixed. Note however that most algorithms for the MAP problem are able to provide a configuration $\mathbf{m}^*$ without resorting to the procedure described (and with much less overhead).

MAP was shown to be NP-hard to approximate even in polytree-shaped networks [30]. Specifically, it was shown that the decision version of MAP is NP^{PP}-complete on loopy networks, and NP-complete on networks of bounded treewidth. More recently, de Campos [3] showed that the problem is NP-hard to solve even in polytree-shaped networks with ternary variables, but admits a fully polynomial-time approximation scheme in networks of bounded treewidth with variables taking on a bounded number of values. A large number of approximate algorithms have been designed to cope with such a computational difficulty, including search-based methods [29], branch-and-bound techniques[17], dynamic programming [12,21], message passing [18,13], function approximation [8,4,26], and knowledge compilation [11].

4 Influence Diagrams and the MEU Problem

An influence diagram extends a Bayesian network with preferences and actions in order to represent decision making situations. Formally, a influence diagram consists of a DAG over a set of *chance variables* $\mathbf{C}$, *decision variables* $\mathbf{D}$, and *value variables* $\mathbf{V}$. The sets $\mathbf{C}$, $\mathbf{D}$ and $\mathbf{V}$ are disjoint. A chance variable C represents quantities over which the decision maker has no control, and is associated with conditional probability assessments $P(C|\text{pa}(C))$ as in a Bayesian network. The restriction of an influence diagram to its chance variables characterizes a Bayesian network. A decision variable D represents possible actions available to the decision maker conditional on the observation of the values of $\text{pa}(D)$. Decision variables are not (initially) associated to any function. A value variable V represents costs or rewards of taking actions $\text{pa}(V) \cap \mathbf{D}$ given an instantiation of $\text{pa}(V) \cap \mathbf{C}$. Every value variable V is associated with utility functions $U(\text{pa}(V)))$, which encode additive terms of the overall utility. The treewidth of an influence diagram is the treewidth of the corresponding moral graph after deleting value nodes.

A *decision rule* (a.k.a policy) for a decision variable D is a conditional distribution $P(D|\text{pa}(D))$ specifying the probability of executing action $D = d$ upon observing $\text{pa}(D) = \pi$. A decision rule prescribes an agent behavior, which is not necessarily deterministic (i.e., the agent might take different actions d when in scenario π according to $P(D|\pi)$). When $P(D|\text{pa}(D))$ is degenerate for every π, we can identify a policy with a function mapping configurations π into actions d. Moreover, if D has no parents, then we can associated a decision rule for D with an assignment of a value of D. We will often refer to degenerate policies as functions or assignments (of root decision variables). A *strategy* is a set containing exactly one decision rule for each decision variable. The expected utility of a strategy $\mathcal{S} = \{P(D|\text{pa}(D)) : D \in \mathbf{D}\}$ is given by

$$E(\mathcal{S}) = \sum_{V \in \mathbf{V}} \sum_{\text{pa}(V)} U(\text{pa}(V)) P(\text{pa}(V)|\mathcal{S}) \tag{2}$$

$$= \sum_{V \in \mathbf{V}} \sum_{\mathbf{C},\mathbf{D}} U(\text{pa}(V)) \prod_{X \in \mathbf{C} \cup \mathbf{D}} P(X|\text{pa}(X)) \,. \tag{3}$$

The MEU problem is to compute the *value* of the maximum expected utility of a strategy, that is, to compute $\max_{\mathcal{S}} E(\mathcal{S})$. A strategy $\mathcal{S}^*$ whose expected utility equals that value is called an *optimal strategy*. We can obtain an optimal strategy by recursively solving MEU problems, in a similar fashion to the computation of a maximum a posteriori configuration of the MAP problem. Although in this work we state the results of the reductions in terms of the MAP and MEU problems (hence, problems whose output are numbers), the same results could be stated with respect to maximum a posteriori configurations of MAP and optimal strategies of MEU. It is well-known that the maximum expected utility can be attained by a strategy containing only degenerate policies. Hence, in what concerns the MEU problem there is no loss in allowing only deterministic policies.

The *perfect recall* condition characterizes a non-forgetting agent, and translates graphically to the property that the parents of any decision variable are also parents of any of its children. Perfect recall is a consequence of rationality when the decision problem involves a single agent with unlimited resources, as it equates with every known information being considered when making a decision. This is not the case when multiple agents are involved or resources such as memory and computing power are limited. A related concept is that of *regularity*, which requires a temporal order over the decision variables. Together, perfect recall and regularity enable the solution of MEU by dynamic programming due to Bellman's principle of optimality. In our definition, we do not require or assume perfect recall or regularity, although we do allow them to be present by explicit specification in the graph. Influence diagrams that do not enforce perfect recall and regularity are often called *limited memory influence diagrams* [16] or *decision networks* [33], although there is some ambiguity about the use of the latter.

De Campos and Ji [5] showed that the decision version of MEU is NP^{PP}-complete in loopy diagrams, and NP-complete in diagrams of bounded treewidth. Mauá et al. strengthened those results by showing the problem to be NP-hard even in polytree-shaped diagrams with ternary variables and a single value variable [25], and even in polytree-shaped diagrams with binary variables and arbitrarily many value variables [23]. They also showed that it is NP-hard to approximately solve the problem, even in polytree-shaped diagrams when variables can take on arbitrarily many values [25], but that there is a fully polynomial-time approximation scheme when both the diagram's treewidth and the maximum variable cardinality are bounded [22]. The problem was also shown to be polynomial-time computable in polytree-shaped diagrams with binary variables [23], and in diagrams that satisfy perfect recall and whose minimal diagram has bounded treewidth [16]. As with MAP, the computational hardness of the problem motivated the development of a large number of approximate algorithms.

Some of the approaches include branch-and-bound [28,27,32,14], dynamic programming [20,12,24], integer programming [5], message passing [19], combinatorial search [16,9], and function approximation [7,6].

5 Reducing MAP To MEU

Consider a MAP problem with Bayesian network $\mathcal{N} = (G, \mathbf{X}, \{P(X|\text{pa}(X)\})$, MAP variables $\mathbf{M} \subseteq \mathbf{X}$ and evidence $\mathbf{E} = \hat{\mathbf{e}}$. Assume w.l.o.g. that the variables in $\mathbf{E}$ have no children [1]. Consider also an ordering $M_1, \ldots, M_n$ of the variables in $\mathbf{M}$ consistent with the partial ordering defined by G (i.e., if there is a directed path from M_i to M_j in G then $j > i$), and an ordering $E_1, \ldots, E_m$ of the variables in $\mathbf{E}$ also consistent with G. Let $\hat{e}_i$ denote the assignment in $\mathbf{e}$ corresponding to E_j, $j = 1, \ldots, m$. Obtain an influence diagram $\mathcal{I}$ by augmenting the Bayesian network $\mathcal{N}$ in the following way.

1. Label every variable in $\mathbf{X}$ as chance variable;
2. Add root chance variables S_0 and T_0 with values t and f, and specify $P(S_0) = P(T_0) = 1/2$;
3. For $i = 1, \ldots, n$ add a decision variable D_i taking the same values as M_i;
4. For $i = 1, \ldots, n$ add a chance variable S_i with values t and f, and parents S_{i-1}, M_i and D_i, and specify

$$P(S_i = 1 | S_{i-1}, M_i, D_i) = \begin{cases} 1, & \text{if } S_{i-1} = t \text{ and } M_i = D_i\,; \\ 0, & \text{otherwise.} \end{cases}$$

5. For $j = 1, \ldots, m$ add a variable T_j with values t and f, parents T_{j-1} and E_j, and specify

$$P(T_j = 1 | T_{j-1}, E_j) = \begin{cases} 1, & \text{if } T_{j-1} = t \text{ and } E_j = \hat{e}_j\,; \\ 0, & \text{otherwise.} \end{cases}$$

6. Add a value variable V with parents S_n and T_m and utility function

$$U(S_n, T_m) = \begin{cases} 1, & \text{if } S_n = T_m = t\,; \\ 0, & \text{otherwise.} \end{cases}$$

Figure 1 illustrates the influence diagram obtained in reduction above.

Remark 1. The above reduction takes time polynomial in the size of the input Bayesian network.

Remark 2. The reduction might introduce (undirected) loops, that is, the reduction (potentially) maps a polytree-shaped Bayesian network into a loopy influence diagram.

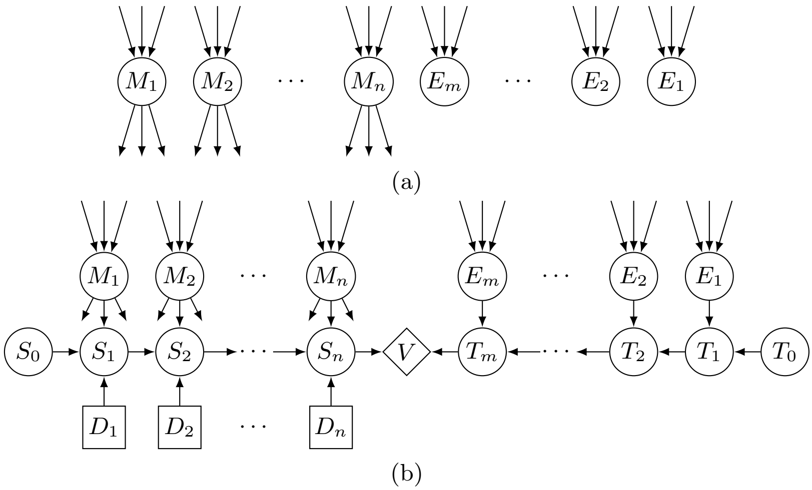

Fig. 1. Fragments of (a) of a Bayesian network and (b) its equivalent influence diagram produced by the procedure described

Lemma 1. *Let P be the probability measure induced by $\mathcal{I}$. Then,*

$$P(S_n = t, S_0, \dots, S_{n-1}|\mathbf{M}, \mathbf{D}) = \begin{cases} 1/2, & \textit{if } S_i = t \textit{ and } M_i = D_i, i = 1, \dots, n\,; \\ 0, & \textit{otherwise.} \end{cases}$$

Proof. By induction in n. The base case for $n = 1$ follows from simple application of the Chain Rule in Bayesian networks:

$$P(S_1 = t, S_0|M_1, D_1) = P(S_0)P(S_1 = t|M_1, D_1) = \begin{cases} 1/2, & \text{if } M_1 = D_1; \\ 0, & \text{otherwise.} \end{cases}$$

Assume the result holds for some n. Applying the Chain Rule and using the conditional independences represented by the graph we obtain

$$P(S_{n+1} = t, S_0, \dots, S_n|M_1, \dots, M_{n+1}, D_1, \dots, D_{n+1}) = \\ P(S_{n+1} = t|S_n, M_{n+1}, D_{n+1})P(S_0, \dots, S_n|M_1, \dots, M_n, D_1, \dots, D_n)\,.$$

By design $P(S_{n+1} = t|S_n, M_{n+1}, D_{n+1})$ vanishes unless $S_n = t$ and $M_{n+1} = D_{n+1}$, in which case the above equality equals

$$P(S_n = t, S_0, \dots, S_{n-1}|M_1, \dots, M_n, D_1, \dots, D_n)\,.$$

Hence, the induction hypothesis holds also for $n + 1$, and the result follows. □

Lemma 2. *Let P be the probability measure induced by $\mathcal{I}$. Then,*

$$P(T_m = t, T_1, \dots, T_{m-1}|\mathbf{E}) = \begin{cases} 1, & \textit{if } T_j = t \textit{ and } E_j = \hat{e}_j, j = 1, \dots, m\,; \\ 0, & \textit{otherwise.} \end{cases}$$

Proof. The proof is similar to the proof of Lemma 1. □

Theorem 1. *The MEU problem obtains the same value as the MAP problem.*

Proof. Consider an arbitrary strategy $\mathcal{S} = \{d_1, \ldots, d_n\}$, and let $\mathbf{S} = \{S_0, \ldots, S_{n-1}\}$ and $\mathbf{T} = \{T_0, \ldots, T_{m-1}\}$. By using the independences stated in $\mathcal{I}$ and the Total and Chain Rules we derive

$$\begin{aligned} E(\mathcal{S}) &= \sum_{S_n, T_m} U(S_n, T_m) P(S_n, T_m | \mathcal{S}) = P(S_n = t, T_m = t | \mathcal{S}) \\ &= \sum_{\mathbf{X}, \mathbf{S}, \mathbf{T}} P(S_n = t, T_m = t, \mathbf{X}, \mathbf{S}, \mathbf{T} | \mathbf{D} = \mathcal{S}) \\ &= \sum_{\mathbf{X}, \mathbf{S}, \mathbf{T}} P(S_n = t, \mathbf{S} | \mathbf{M}, \mathbf{D} = \mathcal{S}) P(T_m = t, \mathbf{T} | \mathbf{E}) P(\mathbf{X}) . \end{aligned}$$

According to Lemmas 1 and 2, the product in the sum above vanishes whenever $S_i \neq t$, for some $i = 1, \ldots, n-1$, $M_i \neq d_i$, for some $i = 1, \ldots, n$, $T_i \neq t$, for $j = 1, \ldots, n-1$, or $E_j \neq \hat{e}_j$, for $j = 1, \ldots, m$. Whence,

$$\begin{aligned} E(\mathcal{S}) &= \sum_{\mathbf{H}, \mathbf{S_0}, \mathbf{T_0}} P(S_n = t | \mathbf{M} = \mathcal{S}) P(T_m = t | \mathbf{E} = \hat{\mathbf{e}}) P(\mathbf{M} = \mathcal{S}, \mathbf{E} = \hat{\mathbf{e}}, \mathbf{H}) P(S_0) P(T_0) \\ &= P(\mathbf{M} = \mathcal{S}, \mathbf{E} = \hat{\mathbf{e}}, \mathbf{H}) . \end{aligned}$$

It follows from the above that $\max_{\mathcal{S}} E(\mathcal{S}) = \max_{\mathbf{h}} P(\mathbf{M} = \mathbf{h}, \mathbf{E} = \hat{\mathbf{e}}, \mathbf{H})$, which proves the result. □

The next result shows that the reduction devised maintains at least part of the structure of the reduced problem.

Theorem 2. *Let κ denote the treewidth of the Bayesian network $\mathcal{N}$. Then the diagram $\mathcal{I}$ has treewidth at most $\kappa + 4$.*

Proof. Let $\mathcal{T}$ be an optimal tree decomposition (i.e., one with minimum treewidth) for the DAG of $\mathcal{N}$ after deleting the arcs leaving variables in $\mathbf{E}$ (removing arcs leaving evidence nodes does not alter the result of MAP inference [1]). We obtain a tree decomposition for $\mathcal{I}$ whose treewidth is at most the treewidth of $\mathcal{T}$ plus three as follows. For $i = 1, \ldots, n$ find a node whose associated variable set includes $\{M_i\} \cup \mathrm{pa}(M_i)$, add a leaf node ℓ_i as its neighbor and associate ℓ_i with $\{M_i\} \cup \mathrm{pa}(M_i)$. Similarly, for $j = 1, \ldots, m$ find a node associated with a superset of $\{E_j\} \cup \mathrm{pa}(E_j)$, add a leaf node ℓ_{n+j} as its neighbor, and associate ℓ_{n+j} with $\{E_j\} \cup \mathrm{pa}(E_j)$. Transform the resulting structure such that it becomes binary, and denote the result by $\mathcal{T}_1$.[3] Root $\mathcal{T}_1$ at a node r (by orienting arcs away from r) such that $\ell_1, \ldots, \ell_{n+m}$ are visited in-order, that is, in a depth-first tree traversal of $\mathcal{T}_1$ rooted at r, ℓ_i is visited before ℓ_j if

[3] Any tree decomposition can be turned into a binary tree decomposition (i.e., one in which each node has at most three neighbors) of same treewidth [31].

and only if $i < j$. Obtain a structure $\mathcal{T}_2$ from $\mathcal{T}_1$ as follows. For every node ℓ_i, $i = 1, \ldots, n$, add a node ℓ'_i as a child of ℓ_i and associate it to $\{S_i, S_{i-1}, D_i, M_i\}$. Similarly, for every node ℓ_i, $i = n+1, \ldots, m$, add a child node ℓ'_i associated to $\{T_i, T_{i-1}, E_i\}$. The structure $\mathcal{T}_3$ is a not a valid tree-decomposition, as it violates the running intersection property: e.g. the variable set associated to a node ℓ'_i, with $i = 1, \ldots, n$, contains the variable S_i, which is also in the variable set associated to ℓ'_{i+1} but not in the variable set associated to any other node in the path between them (as S_i does not appear in $\mathcal{T}$). We obtain a valid tree-decomposition $\mathcal{T}_3$ from $\mathcal{T}_2$ by walking around $\mathcal{T}_2$ in a Euler tour tree traversal where each edge is visited exactly twice and enforcing the running intersection property: for each node that appears after ℓ'_{i-1} and before ℓ'_i during the walk, we include S_{i-1} if $i < n$ and T_{i-1} if $i > n$ in its associated variable set. Since the Euler tour tree traversal visits each leaf once and each internal node at most three times, the procedure inserts at most three new variables in any sets associated to a node of $\mathcal{T}_3$. The treewidth of $\mathcal{T}_3$ thus exceeds the treewidth of $\mathcal{T}_2$ by at most 3. The last step is to obtain $\mathcal{T}'$ from $\mathcal{T}_3$ by covering $\text{pa}(V) = \{S_n, E_m\}$ while respecting the running intersection property. To this end, we include E_m in the variable set associate with every node in the path from ℓ'_n to ℓ'_m. This increases the treewidth by at most one, and guarantees that the treewidth of $\mathcal{T}'$ is in the worst case the treewidth of $\mathcal{T}$ plus four. □

The above result implies that applying the reduction on the class of bounded treewidth Bayesian networks produces a class of bounded treewidth influence diagrams. Thus, (the decision version of) MAP problems that are NP-complete are mapped into (the decision version of) MEU problems which are also NP-complete.

6 Reducing MEU To MAP

Consider a MEU problem with influence diagram $\mathcal{I}$. In order to obtain a Bayesian network $\mathcal{N}$ we first need to apply a sequence of transformations that obtains an MEU-equivalent influence diagram where decision variables have no parents and there is a single value variable. The following transformation substitutes a decision variable with multiple parents by multiple parentless decision variables and preserves the value of the MEU.

Transformation 1 *Select a decision variable D with at least one parent, and let $\pi_1, \ldots, \pi_r$ be the configurations of pa(D).*

1. *Remove D;*
2. *Add parentless decision variables $D_1, \ldots, D_r$ taking the same values as D;*
3. *Add variables $X_1, \ldots, X_r$ taking the same values as D; set pa$(X_1) =$ pa$(D) \cup \{D_1\}$ and pa$(X_i) =$ pa$(D) \cup \{M_i, X_{i-1}\}$ for $i = 2, \ldots, r$; specify*

$$\Pr(X_1|D_1, pa(D)) = \begin{cases} 1, & \text{if } pa(D) = \pi_1 \text{ and } X_1 = D_1\,, \\ 0, & \text{if } pa(D) = \pi_1 \text{ and } X_1 \neq D_1\,, \\ 1/m & \text{if } pa(D) \neq \pi_1\,; \end{cases}$$

for $i = 2, \ldots, r$, *specify*

$$\Pr(X_i|X_{i-1}, D_i, pa(D)) = \begin{cases} 1, & \textit{if } pa(D) \neq \pi_i \textit{ and } X_i = X_{i-1}\,, \\ 0, & \textit{if } pa(D) \neq \pi_i \textit{ and } X_i \neq X_{i-1}\,, \\ 1, & \textit{if } pa(D) = \pi_i \textit{ and } X_i = D_i\,, \\ 0, & \textit{if } pa(D) = \pi_k \textit{ and } X_i \neq D_i\,; \end{cases}$$

4. *Substitute* D *by* X_r *in* $pa(C)$ *for every* C *in* $ch(D)$*, and modify the conditional probability functions* $\Pr(C|pa(C))$ *accordingly.*

Figure 2 depicts the result of applying Transformation 1 on a decision node. The bottleneck of the computational performance of the transformation is the specification of the $O(r^2v^3)$ probability values $\Pr(X_i = x_i|X_{i-1} = x_{i-1}, D_i = d_i, \text{pa}(D) = \pi_k)$, where v is the cardinality of D.

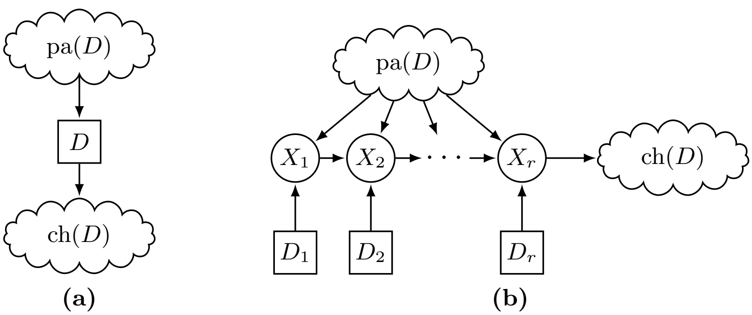

Fig. 2. A piece of a diagram before (a) and after (b) Transformation 1

Remark 3. Let c be the maximum cardinality of a variable in the family of D and $w = |\text{pa}(D)|$. Then the transformation takes time $O(c^{2w+3})$. If we assume that the in-degree of decision variables are bounded, then w is a constant and the transformation takes time polynomial in the input size.

If the in-degree of decision variables is not bounded then the specification of an optimal strategy might take space exponential in the input. Thus, assuming that w is bounded is reasonable.

Remark 4. The transformation might create loops in polytree-shaped diagrams.

The following two results were proved in [25, Proposition 7].

Lemma 3. *Let* $\mathcal{I}'$ *be the result of applying Transformation 1 on a decision variable* D *in a diagram* $\mathcal{I}$*. There is a polynomial-time computable bijection between strategies of* $\mathcal{I}$ *and* $\mathcal{I}'$ *that preserves expected utility.*

Corollary 1. *Let $\mathcal{I}'$ be the result of applying Transformation 1 on a decision variable D in a diagram $\mathcal{I}$. The MEU of $\mathcal{I}'$ and $\mathcal{I}$ are equal.*

Transformation 1 might increase the treewidth of the graph. To see this, consider an influence diagram $\mathcal{I}$ containing one chance variable C, one decision variable D and one value variable V, with graph structure $C \to D \to V$. The treewidth of the transformed diagram is three while the treewidth of original graph is one. The following result shows that the increase in treewidth is small.

Lemma 4. *Transformation 1 increases the treewidth by at most two.*

Proof. Let $\mathcal{I}'$ be the result of applying the transformation in a diagram $\mathcal{I}$ of treewidth κ. Also, let M and M' be the moral graphs of $\mathcal{I}$ and $\mathcal{I}'$, respectively. We can obtain M from M' by sequentially eliminating nodes $D_1, \ldots, D_r$ and $X_1, \ldots, X_{r-1}$, in this order, and replacing X_r with D. Let $M_1, \ldots, M_{2r}$ be the graphs obtained by applying each of these operations. Thus, M_1 is the graph obtained by removing D_1 from M', and M_{2r} equals M. Let $\kappa_1, \ldots, \kappa_{2r}$ be the treewidth of the graphs $M_1, \ldots, M_{2r}$, respectively, and κ' be the treewidth of M'. The node D_1 is simplicial and has degree $|\text{pa}(D)| + 1$ in M'. Since M_1 contains the clique $\{X_r, X_{r-1}, D_r\} \cup \text{pa}(D)$ (where $\text{pa}(D)$ is taken with respect to M), it follows that $\kappa_1 \geq |\text{pa}(X_r)| = |\text{pa}(D)| + 2$, which implies $\kappa = \max\{|\text{pa}(D)| + 1, \kappa_1\} = \kappa_1$. Assume that $\kappa_\ell = \kappa_0$, for some $1 \leq \ell < r - 1$. The variable $D_{\ell+1}$ is simplicial and has degree $|\text{pa}(D)| + 2$ in M_ℓ. The treewidth $\kappa_{\ell+1} \geq |\text{pa}(D)| + 2$ because $M_{\ell+1}$ contains the clique $\{X_r, X_{r-1}, D_r\} \cup \text{pa}(D)$. Hence, $\kappa_\ell = \max\{|\text{pa}(D)| + 2, \kappa_{\ell+1}\} = \kappa_{\ell+1}$, and by induction we have that $\kappa' = \kappa_{r-1}$. The node D_r is simplicial and has degree $|\text{pa}(D)|+2$ in M_{r-1}. Since M_r contains the clique $\{X_r, X_{r-1}\} \cup \text{pa}(D)$, it follows that $\kappa_r \geq |\text{pa}(D)|+1$, and thus $\kappa_{r-1} = \max\{|\text{pa}(D)|+2, \kappa_r\} \leq \kappa_r + 1$. Hence, $\kappa_{r-1} \leq \kappa_r + 1$. A similar reasoning applies for κ_ℓ with $r < \ell < 2r$. M_{r+1} (i.e., the graph obtained by removing X_1) contains a clique of size $|\{X_r, X_{r-1}\} \cup \text{pa}(D)| = |\text{pa}(D)|+2$, and the node X_1 is simplicial and has degree $|\text{pa}(D)| + 1$ in M_r. Hence, $\kappa_r = \max\{|\text{pa}(D)| + 1, \kappa_{r+1}\} = \kappa_{r+1}$. Assume $\kappa_\ell = \kappa_m$ for $r < \ell < 2r - 2$. Then $X_{\ell-m+1}$ is simplicial and has degree $|\text{pa}(D)|+1$ in M_ℓ. Since $M_{\ell+1}$ contains the clique $\{X_r, X_{r-1}\} \cup \text{pa}(D)$, it follows that $\kappa_\ell = \max\{|\text{pa}(D)| + 1, \kappa_{\ell+1}\} = \kappa_{\ell+1}$. Thus, by induction, $\kappa_r = \kappa_{2r-2}$. Finally, the graph M_{2r-1} (obtained by removing X_{r-1} from M_{2r-2}) contains the clique $\{X_r\} \cup \text{pa}(D)$ (so that $\kappa_{2r-1} \geq |\text{pa}(D)|$), and X_{r-1} is simplicial and has degree $|\text{pa}(D)| + 1$ in M_{2r-2}. Therefore, $\kappa_{2r-2} = \max\{|\text{pa}(D)| + 1, k_{2r-1}\} \leq k_{2r-1} + 1$. Since the replacement of X_r with D used to generate $M_{2r} = M$ from M_{2r-1} does not change the treewidth (i.e., $\kappa = \kappa_{2r} = \kappa_{2r-1}$), we have that

$$\kappa' = \kappa_{r-1} \leq \kappa_r + 1 = \kappa_{2r-2} + 1 \leq \kappa_{2r-1} + 2 = \kappa + 2\,,$$

and the result follows. □

The previous result can be generalized to recurrent applications of Transformation 1:

Corollary 2. *Let $\mathcal{I}'$ be the result of applying Transformation 1 in a diagram $\mathcal{I}$ of treewidth κ repeatedly until all decision variables are parentless. Then the treewidth of $\mathcal{I}'$ is at most $\kappa + 2$.*

Proof. Applying the transformation on two different decision variables affect different parts of the graph of the original diagram. Hence, the new variables introduced by the repeated applications can be eliminated in parallel, which shows that the increase in treewidth remains bounded by two. □

A final issue to circumvent in order to devise a mapping from MEU to MAP problems is the treatment of multiple value variables. The following transformation maps diagrams with multiple value variables into MEU-equivalent diagrams with a single value variable.

Transformation 2 *Take an influence diagram $\mathcal{I}$ with value variables $V_1, \ldots, V_n$, and let $\underline{U} = \min_{i,\pi_i} U(pa(V_i) = \pi_i)$ and $\overline{U} = \max_{i,\pi_i} U(pa(V_i) = \pi_i)$ denote, respectively, the minimum and maximum utility value associated to any value variable.*

1. *Substitute each value variables V_i by a binary chance variable W_i taking values t and f and with probability distribution given by*
$$P(W_i\!=\!t|pa(V_i)) = \frac{U(pa(V_i)) - \underline{U}}{\overline{U} - \underline{U}} .$$
2. *Add variables $O_1, \ldots, O_n$, each taking values t and f, with $pa(O_1) = \{W_1\}$, and $pa(O_i) = \{O_{i-1}, W_i\}$, $i = 2, \ldots, n$; specify $P(O_1 = t|W_1 = t) = 1$, $P(O_1\!=\!t|W_1\!=\!f) = 0$ and*
$$P(O_i\!=\!t|O_{i-1}, W_i) = \begin{cases} 1, & \text{if } O_{i-1} = W_i = t \\ (i-1)/i, & \text{if } O_{i-1} = t \text{ and } W_i = f \\ 1/i, & \text{if } O_{i-1} = f \text{ and } W_i = t \\ 0, & \text{if } O_{i-1} = W_i = f \end{cases} ;$$
3. *Add a value variable V with $pa(V) = \{O_n\}$, $U(pa(V) = t) = n\overline{U}$ and $U(pa(V)\!=\!f) = n\underline{U}$.*

Figure 3 illustrates the application of Transformation 2.

Remark 5. The transformation takes time polynomial in the size of the input influence diagram.

Remark 6. The transformation might introduce loops.

The following three results were proved in [22, Theorem 1].

Lemma 5. *Let $\mathcal{I}'$ be the result of applying Transformation 2 on an influence diagram $\mathcal{I}$. There is a polynomial-time computable bijection between strategies of $\mathcal{I}$ and $\mathcal{I}'$ that preserves expected utility.*

Corollary 3. *Let $\mathcal{I}'$ be the result of applying Transformation 2 on an influence diagram $\mathcal{I}$. The MEU of $\mathcal{I}'$ and $\mathcal{I}$ are equal.*

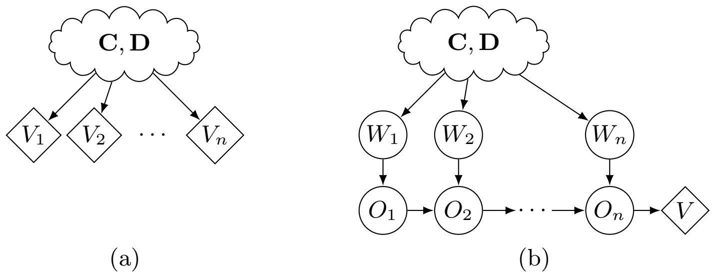

Fig. 3. (a) Influence diagram with multiple value variables. (b) Its equivalent influence diagram obtained by Transformation 2.

Lemma 6. *Transformation 2 increases the treewidth by at most three.*

We are now ready to describe the reduction from MEU to MAP problems.

1. While there is a decision variable with at least one parent, apply Transformation 1;
2. If there is more than a value variable, apply Transformation 2;
3. Transform each (parentless) decision variable D into a chance variable M taking on the same values, and with $P(M) = 1/v$, where v is the cardinality of D;
4. Replace the (single) value variable V by a chance variable E taking values t and f, and with

$$P(E = t|\text{pa}(V)) = \frac{U(\text{pa}(V)) - \underline{U}}{\overline{U} - \underline{U}},$$

where $\underline{U} = \min_\pi U(\text{pa}(V){=}\pi)$ and $\overline{U} = \max_\pi U(\text{pa}(V){=}\pi)$.

Let $\mathcal{N}$ be the Bayesian network obtained by the reduction above, and denote by $\mathbf{M}$ the set of variables introduced in step 3.

Theorem 3. *Let* MAP *be the value of the MAP problem with Bayesian network* $\mathcal{N}$*, MAP variables* $\mathbf{M}$ *and evidence* $E = t$*, and* MEU *be the value of the MEU problem with input* $\mathcal{I}$*. For any configuration* $\mathbf{m}$ *of* $\mathbf{M}$ *we have that*

$$\mathsf{MAP} = \frac{P(\mathbf{M}{=}\mathbf{m})}{\overline{U} - \underline{U}}\mathsf{MEU} - \underline{U},$$

where $P(\mathbf{M}) = \prod_{M \in \mathbf{M}} P(M)$*, and* $\underline{U}$ *and* $\overline{U}$ *are, respectively, the minimum and the maximum of the utility function defined by* $\mathcal{I}$*.*

Proof. Let $\mathbf{X}$ denote the variables in $\mathcal{N}$, and $\mathbf{Y} = \mathbf{X} \setminus (\mathbf{M} \cup \{E\})$. Since the set $\mathbf{M}$ contains only root variables associated to uniform probability distributions, $P(\mathbf{M}{=}\mathbf{m})$ equals some constant C for any configuration $\mathbf{m}$. Hence,

$$\begin{aligned}\mathsf{MAP} &= \max_{\mathbf{m}} \sum_{\mathbf{Y}} P(E{=}t|\text{pa}(E))P(\mathbf{Y}|\mathbf{M}{=}\mathbf{m})P(\mathbf{M}{=}\mathbf{m}) \\ &= C \max_{\mathbf{m}} \sum_{\mathbf{Y}} P(E{=}t|\text{pa}(E))P(\mathbf{Y}|\mathbf{M}{=}\mathbf{m}) \\ &= C \max_{\mathbf{m}} \sum_{\mathbf{Y}} \frac{U(\text{pa}(V)) - \underline{U}}{\overline{U} - \underline{U}} P(\mathbf{Y}|\mathbf{M}{=}\mathbf{m}) = \frac{C}{\overline{U} - \underline{U}} \mathsf{MEU} - \underline{U},\end{aligned}$$

which proves the result. □

The following result shows that the reduction maps NP-complete instances of MEU into NP-complete instances of MAP.

Corollary 4. *Let κ denote the treewidth of an influence diagram. Then the Bayesian network generated by the reduction has treewidth at most $\kappa + 5$.*

Proof. It follows from Lemmas 4 and 6. □

7 Conclusions

Computing the maximum posterior probability of a subset of variables in a Bayesian network and calculating the maximum expected utility of strategies in an influence diagrams are common tasks in probabilistic reasoning. Despite their similarities, these two problems have hitherto been investigated independently. In this work, we showed constructively that these two problems are computationally equivalent in that one problem can be reduced to the other in polynomial time. Hence, any algorithm designed for one problem can be immediately used for the other with a small overhead. Future work should evaluate the benefits and drawbacks of applying algorithms designed for one problem to solve the other, by means of the reductions presented here.

A common limitation of the correspondences devised here is that they map problems with polytree-shaped graph structure into problems with loopy graph structure. This reduces some tractable instances of one problem into apparently intractable instances of the other problem. For instance, MEU is tractable in polytree-shaped diagrams with binary variables and a single value node, but the reduction shown here creates a MAP problem in a loopy Bayesian network, for which no efficient algorithm exists. A similar problem appears if we try to use the reductions developed here to prove the hardness of instances with simple structure. For instance, the complexity of MAP in tree-shaped Bayesian networks with binary variables is not known, and it cannot be characterized by the reduction from MAP to MEU presented here because tree-shaped Bayesian networks are mapped into loopy influence diagrams. It would be interesting to devise reductions that preserve the topology of the graph structure.

Acknowledgments. The author thanks Fabio G. Cozman and the reviewers for their valuable comments and suggestions. This work was partially supported by the São Paulo Research Foundation (FAPESP) grant no. 2013/23197-4.

References

1. Antonucci, A., Piatti, A.: Modeling unreliable observations in Bayesian networks by credal networks. In: Godo, L., Pugliese, A. (eds.) SUM 2009. LNCS, vol. 5785, pp. 28–39. Springer, Heidelberg (2009)
2. Bodlaender, H., Koster, A., van den Eijkhof, F., van der Gaag, L.: Pre-processing for triangulation of probabilistic networks. In: Proceedings of the 17th Conference on Uncertainty in Artificial Intelligence (UAI), pp. 32–39 (2001)
3. de Campos, C.P.: New complexity results for MAP in Bayesian networks. In: Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI), pp. 2100–2106 (2011)
4. de Campos, L.M., Gámez, J.A., Moral, S.: Partial abductive inference in Bayesian networks by using probability trees. In: 5th International Conference on Enterprise Information Systems (ICEIS), pp. 83–91 (2003)
5. de Campos, C.P., Ji, Q.: Strategy selection in influence diagrams using imprecise probabilities. In: Proceedings of the 24th Conference in Uncertainty in Artificial Intelligence (UAI), pp. 121–128 (2008)
6. de Paz, R.C., Gómez-Olmedo, M., Cano, A.: Approximate inference in influence diagrams using binary trees. In: Proceedings of the 6th European Workshop on Probabilistic Graphical Models (PGM), pp. 43–50 (2012)
7. Dechter, R.: An anytime approximation for optimizing policies under uncertainty. In: Workshop of Decision Theoretic Planning, AIPS (2000)
8. Dechter, R., Rish, I.: Mini-buckets: A general scheme for bounded inference. Journal of the ACM 50(2), 107–153 (2003)
9. Detwarasiti, A., Shachter, R.D.: Influence diagrams for team decision analysis. Decision Analysis 2(4), 207–228 (2005)
10. Howard, R.A., Matheson, J.E.: Influence diagrams. In: Readings on the Principles and Applications of Decision Analysis, pp. 721–762. Strategic Decisions Group (1984)
11. Huang, J., Chavira, M., Darwiche, A.: Solving MAP exactly by searching on compiled arithmetic circuits. In: Proceedings of the 21st National Conference on Artificial Intelligence (NCAI), pp. 1143–1148 (2006)
12. Jensen, F.V., Nielsen, T.D.: Bayesian Networks and Decision Graphs, 2nd edn. Information Science and Statistics. Springer (2007)
13. Jiang, J., Rai, P., Daume III, H.: Message-passing for approximate MAP inference with latent variables. In: Advances in Neural Information Processing Systems 24 (NIPS), pp. 1197–1205 (2011)
14. Khaled, A., Yuan, C., Hansen, E.: Solving limited memory influence diagrams using branch-and-bound search. In: Proceedings of the 29th Conference on Uncertainty in Artificial Intelligence, UAI (2013)
15. Koller, D., Friedman, N.: Probabilistic Graphical Models. MIT Press (2009)
16. Lauritzen, S.L., Nilsson, D.: Representing and solving decision problems with limited information. Management Science 47, 1235–1251 (2001)
17. Lim, H., Yuan, C., Hansen, E.: Scaling up MAP search in Bayesian networks using external memory. In: Proceedings of the 5th European Workshop on Probabilistic Graphical Models (PGM), pp. 177–184 (2010)
18. Liu, Q., Ihler, A.: Variational algorithms for marginal MAP. In: Proceedings of the 27th Conference on Uncertainty in Artificial Intelligence (UAI), pp. 453–462 (2011)

19. Liu, Q., Ihler, A.: Belief propagation for structured decision making. In: Proceedings of the 28th Conference on Uncertainty in Artificial Intelligence (UAI), pp. 523–532 (2012)
20. Madsen, A.L., Nilsson, D.: Solving influence diagrams using HUGIN, Shafer-Shenoy and lazy propagation. In: Proceedings of the 17th Conference in Uncertainty in Artificial Intelligence (UAI), pp. 337–345 (2001)
21. Mauá, D.D., de Campos, C.P.: Anytime marginal MAP inference. In: Proceedings of the 28th International Conference on Machine Learning (ICML), pp. 1471–1478 (2012)
22. Mauá, D.D., de Campos, C.P., Zaffalon, M.: The complexity of approximately solving influence diagrams. In: Proceedings of the 28th Conference on Uncertainty in Artificial Intelligence (UAI), pp. 604–613 (2012)
23. Mauá, D.D., de Campos, C.P., Zaffalon, M.: On the complexity of solving polytree-shaped limited memory influence diagrams with binary variables. Artificial Intelligence 205, 30–38 (2013)
24. Mauá, D.D., de Campos, C.P.: Solving decision problems with limited information. In: Advances in Neural Information Processing Systems 24 (NIPS), pp. 603–611 (2011)
25. Mauá, D.D., de Campos, C.P., Zaffalon, M.: Solving limited memory influence diagrams. Journal of Artificial Intelligence Research 44, 97–140 (2012)
26. Meek, C., Wexler, Y.: Approximating max-sum-product problems using multiplicative error bounds. Bayesian Statistics 9, 439–472 (2011)
27. Nilsson, D., Höhle, M.: Computing bounds on expected utilities for optimal policies based on limited information. Research Report 94, Dina (2001)
28. Nilsson, D., Lauritzen, S.L.: Evaluating influence diagrams using LIMIDs. In: Proceedings of the 16th Conference in Uncertainty in Artificial Intelligence (UAI), pp. 436–445 (2000)
29. Park, J.D., Darwiche, A.: Solving MAP exactly using systematic search. In: Proceedings of the 19th Conference on Uncertainty in Artificial Intelligence (UAI), pp. 459–468 (2003)
30. Park, J.D., Darwiche, A.: Complexity results and approximation strategies for MAP explanations. Journal of Artificial Intelligence Research 21, 101–133 (2004)
31. Shenoy, P.P.: Binary join trees for computing marginals in the Shenoy-Shafer architecture. International Journal of Approximate Reasoning 17(2-3), 239–263 (1997)
32. Yuan, C., Wu, X., Hansen, E.A.: Solving multistage influence diagrams using branch-and-bound search. In: Proceedings of the 26th Conference on Uncertainty in Artificial Intelligence (UAI), pp. 691–700 (2010)
33. Zhang, N.L., Qi, R., Poole, D.: A computational theory of decision networks. International Journal of Approximate Reasoning 11(2), 83–158 (1994)