On the Complexity of Solving Polytree-Shaped Limited Memory Influence Diagrams with Binary Variables

Denis Deratani Mauá[a,*], Cassio Polpo de Campos[a], Marco Zaffalon[a]

[a]*Istituto Dalle Molle di Studi sull'Intelligenza Articiale (IDSIA)*
Galleria 2, Manno, 6928 Switzerland

Abstract

Influence diagrams are intuitive and concise representations of structured decision problems. When the problem is non-Markovian, an optimal strategy can be exponentially large in the size of the diagram. We can avoid the inherent intractability by constraining the size of admissible strategies, giving rise to limited memory influence diagrams. A valuable question is then how small do strategies need to be to enable efficient optimal planning. Arguably, the smallest strategies one can conceive simply prescribe an action for each time step, without considering past decisions or observations. Previous work has shown that finding such optimal strategies even for polytree-shaped diagrams with ternary variables and a single value node is NP-hard, but the case of binary variables was left open. In this paper we address such a case, by first noting that optimal strategies can be obtained in polynomial time for polytree-shaped diagrams with binary variables and a single value node. We then show that the same problem is NP-hard if the diagram has multiple value nodes. These two results close the fixed-parameter complexity analysis of optimal strategy selection in influence diagrams parametrized by the shape of the diagram, the number of value nodes and the maximum variable cardinality.

Keywords: decision theory, influence diagrams, decision networks, probabilistic planning, computational complexity.

1. Introduction

Planning with influence diagrams in partially observable domains suffers from the so-called curse of history: the size of an optimal strategy grows exponentially large with the number of look-ahead steps considered. Lauritzen and Nilsson proposed using limited-memory strategies to avoid the complexity blow up, coining the term *limited memory influence diagrams* (limids) [1]. Limids relax the *no-forgetting* requirement of traditional influence diagrams, and require

---

Table 1: Parametrized complexity of the decision version of the optimal strategy selection problem in limids of bounded treewidth.

topology	num of value nodes	max variable cardinality	complexity
polytree	one	two	P
polytree	unbounded	two	NP-complete
polytree	one	three	NP-complete
polytree	one	unbounded	NP-complete
loopy	one	two	NP-complete
loopy	unbounded	bounded	NP-complete

(implicitly) that the maximum size of an optimal strategy be given as part of the input. In fact, the use of finite-state controllers, which in the influence diagram formalism corresponds to limited memory, has long been seen as a viable alternative in probabilistic planning for long- or infinite-horizon problems, especially when teams are considered [2–6].

In many cases, however, limiting the agent's memory resources does not make the problem tractable, and computing optimal strategies even for structurally very simple limids can be hard. We have shown in a previous work that finding an optimal strategy for polytree-shaped limids is NP-hard even if variables are ternary and there is only a single value node [7]. Also finding a strategy whose expected utility is within a factor of 2^p of the maximum expected utility, where p denotes the number of numerical parameters in the model, is NP-hard, even in polytree-shaped diagrams (with variables taking on arbitrarily many values). On the other hand, we have shown that when the variable cardinalities and the treewidth of the diagram are bounded, a fully polynomial-time approximation scheme exists [8], and that optimal strategies can often be obtained in feasible time [9]. For the remaining cases, efficiency can be achieved at the expense of provably good accuracy by local search methods [1, 4] and message-passing algorithms [10, 11], while branch-and-bound procedures can be used for guaranteed accuracy [12, 13]. Table 1 summarizes the fixed-parameter computational complexity of the decision version of finding optimal strategies in limids of bounded treewidth with respect to topology, number of value nodes and maximum variable cardinality.[1] The first two rows contain novel results presented here. The results in the remaining rows appeared in references [7] and [8].

De Campos and Ji showed that the problem of selecting an optimal strategy in limids with a single value node can be reduced to that of computing a (tight) marginal probability bound in a credal network, and subsequently addressed by

---

[1]The decision version of the optimal strategy selection problem is to decide, for a given a limid, whether there is a strategy whose expected utility exceeds a given rational number.

credal network inference algorithms [12]. The reduction preserves the topology of the graph and the cardinality of the variables; in particular, the reduction maps polytree-shaped diagrams with binary variables into credal polytrees with binary variables [14].

Credal networks are generalizations of Bayesian networks designed to cope with set-valued specifications of the numerical parameters of the model [15]. Computing tight bounds on the marginal probability of some variable in a credal network is NP-hard even when the network is polytree-shaped and variables are ternary [16]. However, Fagiuoli and Zaffalon's 2U algorithm [17] is able to obtain tight marginal bounds in polynomial time in polytree-shaped credal networks with binary variables. Therefore, we can find optimal strategies in polytree-shaped limids with binary chance variables and a single value node by running 2U in the corresponding credal network obtained by de Campos and Ji's reduction. This immediate and important result, which we discuss in Section 3, has apparently not been noticed before in the literature.

When the diagram has multiple value nodes, the transformation of de Campos and Ji reduces the selection of an optimal strategy to the computation of a tight bound for a *sum* of marginals of variables in a credal network, which cannot be solved in polynomial time by standard inference algorithms for credal networks and more specifically by 2U. A possible workaround is to apply a pre-processing step that transforms any diagram into an equivalent diagram that contains a single value node. However, known techniques to map a limid with multiple value nodes into an equivalent limid containing a single value node do not preserve the topology [8]. In particular, they map polytrees into multiply connected graphs. Thus, while the mapping from limids into credal networks enlarges the available toolset of algorithms for solving limids with that of credal network inference, it does not help us in answering the question of whether polytree-shaped diagrams with binary variables and multiple value nodes are in fact NP-hard to solve.

In Section 4, we answer that question affirmatively by a reduction from the partition problem. This result closes the fixed-parameter complexity analysis of the optimal strategy selection problem in limids in what regards the topology of the diagram, the number of values nodes, and the cardinality of the variables: polytree-shaped limids with binary variables and a single value node can be solved in polynomial time, and relaxing any of these assumptions leads to NP-hard problems.

2. Limited Memory Influence Diagrams

To help introduce the concepts and motivate the use of limited memory influence diagrams, consider the following illustrative example of a decision problem under uncertainty.

> Consider a system whose failure depends on statistically independent events $e_1, \ldots, e_n$, which occur with known probabilities $p_1, \ldots, p_n$, respectively. Let E_i $(i = 1, \ldots, n)$ be a binary variable denoting whether
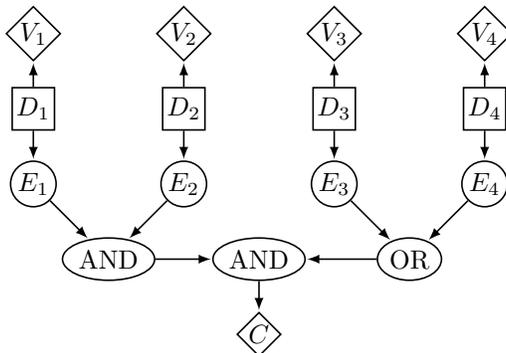
Figure 1: Influence diagram representation of the intervention policy problem described in the text.

e_i occurs ($E_i = 1$ if e_i occurs), and assume that the system failure is determined by a logical function $F(E_i, \ldots, E_n)$ of the causing events ($F = 1$ if the system fails). Suppose that at a given time, the system administrator can intervene to prevent any event e_i from occurring at a cost v_i. Let D_i denote the decision of intervening on event e_i. The conditional probability of the event e_i given an intervention policy for the ith event is given by $P(E_i = 1 | D_i = 1) = 0$ and $P(E_i = 1 | D_i = 0) = p_i$. The cost of a system failure is represented by a variable C. Thus, the expected cost of a combination of intervention policies $\delta_1, \ldots, \delta_n$ is given by

$$\sum_{F, E_1, \ldots, E_n} E[C|F] P(F | E_1, \ldots, E_n) \prod_{i=1}^{n} P(E_i | D_i = \delta_i) + \sum_{i=1}^{n} v_i \delta_i \,.$$

The goal is to select the combination of policies that minimizes the expression above. Figure 1 shows the influence diagram representation of the intervention problem where system failure is determined by $F = (E_1 \wedge E_2) \wedge (E_3 \vee E_4)$.

As in the example above, the quantities in a decision problem can be partitioned into state (or chance) variables $\mathcal{S} = \{S_1, \ldots, S_n\}$, action (or decision) variables $\mathcal{A} = \{A_1, \ldots, A_m\}$, and value variables $\mathcal{V} = \{V_1, \ldots, V_\ell\}$. The state variables represent the unknown quantities over which the agent has no control. The action variables enumerate the alternative courses of action. The value variables assess the quality of decisions for a given state of the world. We assume that variables take on finitely many values.

Influence diagrams are graph-based representations of structured decision problems [18]. An influence diagram represents both the agent's architecture (i.e., what information is available to the agent at each decision stage of the problem) and the environment by means of a directed acyclic graph where each node is associated with exactly one variable of the problem. The nodes in the graph are also partitioned into sets of state, action and value nodes, according to the type of variable with which they are associated, and we refer to nodes

and their associated variables interchangeably. An arc from a node X into an action node A in the graph indicates that a different action $A = a$ can be taken for each possible value of X.

A strategy $\delta = (\delta_1, \ldots, \delta_m)$ is a vector of local decision rules, or policies, one for each action variable in the problem. Each policy δ_j is a mapping from the configurations of the values of the parents $\mathtt{Pa}(A_j)$ of A_j to values of A_j. A policy for an action variable with no parents is simply an assignment of a value to that variable. We assume that policies are encoded as tables. Hence, the size of a policy is exponential in the number of parents of the corresponding action variable, which in real scenarios force us to constraint the maximum number of parents of an action node lest the implementation of a policy be not practicable.

The *no-forgetting* condition (a.k.a. perfect recall) assumes that all decisions and observations are "remembered". Graphically, it entails that if A and A' are two action nodes such that A is a parent of A', then all parents of A are also parents of A'.[2] An influence diagram is said to have limited memory if the no-forgetting condition is not met. The least memory intensive strategy is the one in which actions are taken unconditionally. It is graphically represented by a limid whose action nodes have no parents.

An arc from a node X into a state node S indicates that S is (potentially) stochastically dependent on X. State variables satisfy the Markov condition, which states that any (state) variable is independent of its non-descendant non-parents conditional on its parents. An arc entering a value node V from a node X indicates that the variable V is a (deterministic) function of X. The value variables are assumed to be associated to leaf nodes in the graph, and the overall utility U is assumed to decompose additively in terms of the value variables [19], that is, $U(\mathcal{S}, \mathcal{A}) = V_1(\mathtt{Pa}(V_1)) + \cdots + V_\ell(\mathtt{Pa}(V_\ell))$.

An influence diagram specification consists of its graph, the tabular specification of each value variable as a function $V_k(\mathtt{Pa}(V_k))$ of the values of its parents, and the tabular specification of the conditional probability distributions $P(S_i|\mathtt{Pa}(S_i))$ of each state variable S_i. We assume that any numerical parameter is given as a *rational* number.

Given an action variable A_j and a policy δ_j, we let $P(A_j|\mathtt{Pa}(A_j), \delta_j)$ be the collection of degenerate conditional probability distributions that assign all mass to $a_j = \delta_j(\mathtt{Pa}(A_j))$ (or the degenerate marginal distribution $P(A_j|\delta_j)$ that places all mass on δ_j in case A_j has no parents). With this correspondence between policies and (conditional) probability distributions, we can define a joint probability distribution over the state and action variables for any given strategy δ as

$$P(\mathcal{S}, \mathcal{A}|\delta) = \prod_{i=1}^{n} P(S_i|\mathtt{Pa}(S_i)) \prod_{j=1}^{m} P(A_j|\mathtt{Pa}(A_j), \delta) \,. \tag{1}$$

---

[2]We assume here that when no-forgetting is satisfied the "remembered" arcs are explicitly represented in the diagram.

The expected utility of a strategy δ is then

$$E[U|\delta] = \sum_{k=1}^{\ell} E[V_k|\delta] = \sum_{\mathcal{S},\mathcal{A}} \left(\sum_{k=1}^{\ell} V_k(\mathtt{Pa}(V_k)) \right) P(\mathcal{S},\mathcal{A}|\delta). \qquad (2)$$

The *optimal strategy selection problem* consists in finding a strategy δ^* such that $E[U|\delta^*] \geq E[U|\delta]$ for all δ, and to compute $E[U|\delta^*]$.

2.1. Polytrees, Treewidth and Minimality

The complexity of finding optimal strategies is greatly affected by the shape of the diagram. We say that a limid is *polytree-shaped* if the undirected graph obtained by dropping arc directions is a tree. For example, the limid in Figure 1 is polytree-shaped. A limid that is not polytree-shaped is said to be loopy. An important graph-theoretic measure of the complexity of a limid is its *treewidth*, which measures its resemblance to a(n undirected) tree. Polytree-shaped limids have treewidth given by the maximum in-degree of a node in the graph.

Not all the information in a limid is necessarily relevant to the computation of optimal strategies, and the complexity of the problem can be drastically reduced by removing nodes and arcs that do not affect the expected utility of any strategy. A state or action node is called *barren* if it either has no children or all of its children are barren. Barren nodes have no influence on any value node and thus no impact on the selection of an optimal strategy [20]. Further irrelevances can be found by applying the concept of nonrequisiteness. A parent X of an action node A is *nonrequisite* to A if X is *d*-separated from all the value nodes that descend from A given $\mathtt{Pa}(A) \cup \{A\} \setminus \{X\}$. The arc from X to A is then said to be a *nonrequisite arc*. Nonrequisite arcs indicate that variable X is irrelevant to selecting an optimal policy for A, and its removal leaves the expected utility of the optimal strategy unchanged. A variable that is nonrequisite to all its children can be safely removed from the diagram without affecting the expected utility of an optimal strategy.[3] Thus, the exponential growth of policies can be avoided if all memory arcs (that is, arcs from parents of a parent action node into an action node) are nonrequisite. This is the case, for instance, when state variables form a chain of observable variables in the graph.

We say that a limid is *minimal* if it contains no nonrequisite arcs or barren nodes. Given a limid we can obtain its corresponding minimal form in polynomial time by repeatedly removing nonrequisite arcs and barren nodes [1].

In polytree-shaped diagrams, all arcs entering action nodes are by definition nonrequisite. Hence, minimal polytree-shaped limids are the least memory-intensive architecture, since in this case a strategy simply prescribes a value for every action variable.

---

[3]We use the definition of nonrequisiteness given in Ref. [21] for the single value node case and extended to multiply value nodes in Ref. [1]. Similar (but not completely equivalent) definitions of nonrequisiteness appeared in Refs. [22–24].

3. The 2U Algorithm

A credal network is a concise representation of a set of Bayesian networks, all sharing the same set of variables and graph structure. Formally, a credal network is a directed acyclic graph over a set of state variables, where for each variable and each configuration of its parents there is an associated set of conditional probability distributions. Marginal inference in credal networks consists in computing the upper and lower marginal probabilities of a target variable on the set of Bayesian networks represented by the credal network. The 2U algorithm performs marginal inference in polynomial time in credal polytrees with binary variables.

A limid can be interpreted as a concise description of the set of Bayesian networks that represent the joint distributions $P(\mathcal{S}, \mathcal{A}|\delta)$ induced by strategies δ. Additionaly, as noted by Cooper [25], the expected utility of any strategy δ in a limid with a single value variable V can be reduced to the computation of the marginal probability $P(V'=1)$ in the Bayesian network that represents $P(V', \mathcal{S}, \mathcal{A}|\delta) = P(V'|\text{Pa}(V))P(\mathcal{S}, \mathcal{A}|\delta)$, where V' is a binary state variable with the same parent set of V, and $P(V'=1|\text{Pa}(V)) \propto V(\text{Pa}(V))$. De Campos and Ji's reduction makes use of these two facts to reduce the computation of the maximum expected utility in limids with single value nodes into a marginal inference in a credal network. The reduction leaves state nodes unchanged, converts the value variable V into a state variable V', and turns each action node into a state node associated with the set of policies of the corresponding action variable. The maximum expected utility of a limid is equal to the upper marginal probability of V' according to the corresponding credal network. Hence, we can transform any polytree-shaped limid with binary variables and a single value node into a credal polytree over binary variables, and solve the optimal strategy problem by running 2U in such a credal network.

Instead of performing this two-step procedure, we can adapt 2U to work directly on the limid formalism. The pseudo-code in Algorithm 1 implements 2U in the language of limids. The function $b(y, r)$ in the algorithm is defined for binary y and real r as $b(0, r) = 1 - r$ and $b(1, r) = r$. Given a polytree-shaped limid with binary variables and a single value node, the algorithm returns the maximum expected utility of a strategy. An optimal strategy can be obtained by a small modification to the algorithm that retrieves the arguments of each optimization performed. The algorithm operates only on ancestors of the value variable, as the non-ancestor nodes are barren. The numbers p_X and q_X for state and value nodes can be computed each by enumerating the $2^{|\text{Pa}(X)|}$ combinations of values of r_Y, $Y \in \text{Pa}(X)$, followed by $|\text{Pa}(X)|2^{|\text{Pa}(X)|}$ arithmetic operations, taking a total time of $O(|\text{Pa}(X)|2^{2|\text{Pa}(X)|})$ per state or value node. Since the limid in the input contains the specification of $P(X|\text{Pa}(X))$, which requires at least $2^{|\text{Pa}(X)|}$ numbers, computing each p_X and q_X takes time polynomial in the input size. The total running time of algorithm for a limid with n variables and treewidth w is $O(nw2^{2w})$, which is polynomial in the input size.

Algorithm 1 Pseudo-code of the 2U algorithm adapted for solving limids.

Remove non-ancestors of the value variable
for each variable X in topological order **do**
    if X is a state variable **then**
        $p_X \leftarrow \min\{\sum_{\mathtt{Pa}(X)} P(X\!=\!1|\mathtt{Pa}(X)) \prod_{Y\in\mathtt{Pa}(X)} b(Y, r_Y) : r_Y = p_Y, q_Y\}$
        $q_X \leftarrow \max\{\sum_{\mathtt{Pa}(X)} P(X\!=\!1|\mathtt{Pa}(X)) \prod_{Y\in\mathtt{Pa}(X)} b(Y, r_Y) : r_Y = p_Y, q_Y\}$
    else
        if X is an action variable **then**
            $p_X \leftarrow 0, q_X \leftarrow 1$
        else
            return $\max\{\sum_{\mathtt{Pa}(X)} X(\mathtt{Pa}(X)) \prod_{Y\in\mathtt{Pa}(X)} b(Y, r_Y) : r_Y = p_Y, q_Y\}$
        end if
    end if
end for
---

4. Strategy Selection in Polytree-Shaped Limids with Binary Variables is NP-Hard

In this section we show that the optimal strategy selection problem is NP-hard in polytree-shaped limids with binary variables (and multiple value nodes) by a many-one reduction from the *partition problem*, which can be stated as follows.

> Given positive even integer numbers $z_1, \ldots, z_n$, is it possible to partition them into two sets of equal sum?

The problem is well-known to be NP-complete [26].[4] As usual, we assume that the instances of the partition problem are "reasonably" and "concisely" encoded as bit-strings of length $b = 2(\sum_{i=1}^{n} \lceil \log_2 z_i \rceil + n - 1)$.[5] Any partition of the numbers into two sets can be represented as an n-dimensional binary vector $(\delta_1, \ldots, \delta_n) \in \{0, 1\}^n$. Let $z \overset{\text{def}}{=} \frac{1}{2}\sum_{i=1}^{n} z_i$. The partition problem is equivalent to deciding whether there is a binary vector $(\delta_1, \ldots, \delta_n)$ such that $\sum_{i=1}^{n} z_i \delta_i = \sum_{i=1}^{n} z_i(1 - \delta_i) = z$. A binary vector satisfying that equality is said to be a *yes-solution* of the problem, otherwise it is called a *no-solution*. In either case, the vector is called a solution and the quantity $\sum_{i=1}^{n} z_i \delta_i$ is called its *value*. Since the input numbers are even, also the value of a solution is an even number. Moreover, since only yes-solutions have value z, and z is an integer number, the value of any no-solution $(\delta_1, \ldots, \delta_n)$ satisfies $|z - \sum_{i=1}^{n} z_i \delta_i| \geq 1$.

---

[4]The standard definition of the problem does not require numbers to be even. This constraint however does not alter the complexity of the problem, as an instance of the partition problem with odd numbers admits a yes-solution if and only if the problem instance obtained by doubling each number admits a yes-solution.

[5]The usual encoding of an instance of the partition problem is a binary string $s_1 01 s_2 01 \cdots 01 s_n$, where each substring s_i is the binary representation of the number z_i with every digit duplicated. For example, the encoding of the problem $z_1 = 2$ and $z_2 = 3$ is 1100011111.
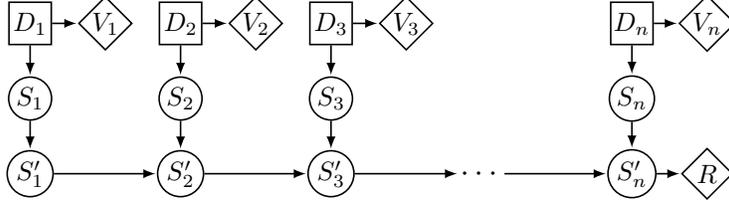
Figure 2: Influence diagram used to solve the partition problem.

To show that the optimal strategy selection problem is NP-hard, we shall design a family of polytree-shaped limids over binary variables such that for each instance $z_1, \ldots, z_n$ of the partition problem there is a limid in the family whose size is polynomial in b (the size of the partition problem) and satisfies $\max_\delta E[U|\delta] \geq 0$ if and only if $\sum_{i=1}^n z_i \delta_i / z = 1$ for some solution δ.

Given an instance of the partition problem we build a limid whose graph structure is shown in Figure 2. For $i = 1, \ldots, n$, the action variable D_i is binary and denotes in which of two sets the number z_i is inserted. Thus, a strategy $\delta = (\delta_1, \ldots, \delta_n) \in \{0,1\}^n$ represents a partition of the input numbers into two sets. The value variables $V_1, \ldots, V_n$ are set so that $V_i \stackrel{\text{def}}{=} -z_i D_i / z$. Hence, $\sum_i E[V_i|\delta] = -\sum_{i=1}^n z_i \delta_i / z$, which equals minus one if and only if the strategy δ is a yes-solution to the partition problem. The state variables are binary and take values in $\{0,1\}$. For $i = 1, \ldots, n$, the conditional probabilities of S_i given its parent D_i are specified such that $P(S_i{=}1|D_i{=}0) \stackrel{\text{def}}{=} 1$ and

$$P(S_i{=}1|D_i{=}1) \stackrel{\text{def}}{=} \min\left\{1, \sum_{k=0}^{12b}(-1)^k \frac{[z_i/z]^k}{k!}\right\}.$$

The second argument in the minimization above is, by Lemma 4 in the appendix, always positive, and can be computed by $O(12b)$ arithmetic operations (used to obtain the terms $[z_i/z]^k$), plus $O(12b)$ operations that compute the factorials $k!$, $k = 0, \ldots, 12b$. Hence, the right-hand side of the identity above is a number between zero and one computed in time polynomial in b. The variables $S_1', \ldots, S_n'$ are deterministic, and satisfy $S_1' \stackrel{\text{def}}{=} S_1$ and $S_i' \stackrel{\text{def}}{=} S_{i-1}' S_i$, $i = 2, \ldots, n$. A simple inductive argument on $i = 1, \ldots, n$ shows that $S_n' = S_1 S_2 \cdots S_n$, so that for any strategy δ we have that

$$E[S_n'|\delta] = \sum_{S_1, \ldots, S_n} P(S_n'{=}1|S_1, \ldots, S_n) P(S_1, \ldots, S_n|\delta)$$

$$= \sum_{S_1, \ldots, S_n} S_1 S_2 \cdots S_n \prod_{i=1}^n P(S_i|D_i{=}\delta_i) = \prod_{i=1}^n P(S_i{=}1|D_i{=}1)^{\delta_i}. \quad (3)$$

9

Finally, we specify the value variable R as

$$R \overset{\text{def}}{=} 2^{-9b} + 2 - qS_n', \qquad \text{where} \quad q \overset{\text{def}}{=} \sum_{k=0}^{12b} \frac{1}{k!}.$$

An argument similar to that of the numbers $P(S_i\!=\!1|D_i\!=\!1)$ shows that R is computed in time polynomial in b. The expected utility of any strategy δ is

$$E[U|\delta] = E[R|\delta] + \sum_{i=1}^{n} E[V_i|\delta]$$

$$= 2^{-9b} + 2 - q\prod_{i=1}^{n} P(S_i\!=\!1|D_i\!=\!1)^{\delta_i} - \frac{1}{z}\sum_{i=1}^{n} z_i\delta_i. \qquad (4)$$

To understand the rationale of the reduction, consider the asymptotic expected utility of a strategy as the size of the partition problem goes to infinity:

$$\lim_{b\to\infty} E[U|\delta] = \lim_{b\to\infty} \left(2^{-9b} + 2 - q\prod_{i=1}^{n} P(S_i\!=\!1|D_i\!=\!1)^{\delta_i} - \frac{1}{z}\sum_{i=1}^{n} z_i\delta_i \right)$$

$$= 2 - \left(\lim_{b\to\infty} \sum_{k=0}^{12b} \frac{1}{k!} \right) \prod_{i=1}^{n} \lim_{b\to\infty} \left(\sum_{k=0}^{12b} (-1)^k \frac{[z_i/z]^k}{k!} \right)^{\delta_i} - \frac{1}{z}\sum_{i=1}^{n} z_i\delta_i,$$

which, since $\lim_{b\to\infty} \sum_{k=0}^{12b} (-1)^k [z_i/z]^k/k! = e^{-z_i/z}$ and $\lim_{b\to\infty} \sum_{k=0}^{12b} 1/k! = e$,

$$= 2 - \exp\left(1 - \frac{1}{z}\sum_{i=1}^{n} z_i\delta_i \right) - \frac{1}{z}\sum_{i=1}^{n} z_i\delta_i. \qquad (5)$$

The last expression can be regarded as a function of $s_\delta \overset{\text{def}}{=} \sum_{i=1}^{n} z_i\delta_i/z$, which is a variable that takes values in $[0,2]$ and equals the value of strategy δ up to the factor z. The graph of that expression as a function of a continuous variable s in the range $[0,2]$ is depicted in Figure 3. As we can see from its graph (and more rigorously by analyzing its derivatives), the function is concave, and achieves a maximum value of zero uniquely at $s = 1$. Since $s_\delta = 1$ if and only if the partition problem admits a yes-solution, we can decide a hypothetical instance of the partition problem of infinite size by verifying the sign of the maximum expected utility of a strategy in the corresponding limid. For any finite b, the expected utility of a strategy (given by (4)) differs from (5), and the value of the expected utility of yes-solution depends on the numbers in the partition problem in the input. We shall show that this difference is bounded by a sufficiently small function of the size b of the partition problem, so that yes- and no-solutions can be distinguished in the finite case (i.e., $b < \infty$) by the sign of their corresponding expected utility.

The next lemma shows that for any partition problem the expected utility of a strategy differs from (5) by at most 2^{-9b}.
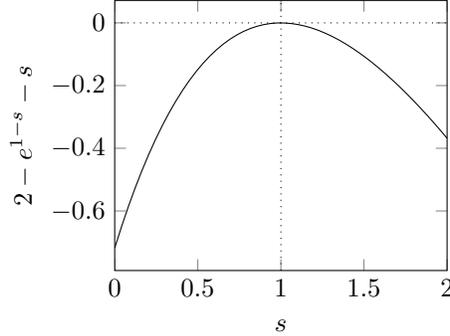
10

Figure 3: Asymptotic expected utility of strategies as $b \to \infty$.

Lemma 1. *Consider the continuous function*

$$f(s) = 2^{-9b} + 2 - \exp(1 - s) - s \,. \tag{6}$$

It follows for any strategy δ that $0 < f\left(\sum_{i=1}^{n} z_i \delta_i / z\right) - E[U|\delta] < 2^{-9b}$.

PROOF. It follows immediately from (4) and (6) that $f\left(\sum_{i=1}^{n} z_i \delta_i / z\right) - E[U|\delta] = qE[S_n'|\delta] - \exp\left(1 - \sum_{i=1}^{n} z_i \delta_i / z\right)$. According to Lemma 4 in the appendix (with $x = z_i / z$ and $M = 12b$), we have for $i = 1, \ldots, n$ that $P(S_i = 1|D_i = 1) = e^{-z_i/z} + \epsilon_i$, where $0 < \epsilon_i < 2^{-12b} < 2^{-2-10b}$. It follows that $E[S_n'|\delta] = \prod_{i=1}^{n}(e^{-z_i/z} + \epsilon_i)^{\delta_i} > \prod_{i=1}^{n}(e^{-z_i/z})^{\delta_i} = \exp\left(-\sum_{i=1}^{n} z_i \delta_i / z\right)$. By an argument similar to Lemma 4, we have that $e < q < e + 2^{-2-9b}$, from which it follows that $qE[S_n'|\delta] > eE[S_n'|\delta] > \exp\left(1 - \sum_{i=1}^{n} z_i \delta_i / z\right)$, and therefore $qE[S_n'|\delta] - \exp\left(1 - \sum_{i=1}^{n} z_i \delta_i / z\right) > 0$.

It remains to show that the upper bound holds. According to the Multivariate Binomial Theorem, we have that

$$E[S_n'|\delta] = \prod_{i=1}^{n}(e^{-z_i/z} + \epsilon_i)^{\delta_i} = \sum_{k \in \mathcal{C}} \prod_{i=1}^{n} e^{-z_i k_i/z} \epsilon_i^{\delta_i - k_i}$$

$$= \exp\left(-\sum_{i=1}^{n} z_i \delta_i / z\right) + \sum_{k \in \mathcal{C}, k \neq \delta} \prod_{i=1}^{n} e^{-z_i k_i/z} \epsilon_i^{\delta_i - k_i} \,,$$

where $\mathcal{C} = \{(k_1, \ldots, k_n) \in \{0,1\}^n : k_i \leq \delta_i, i = 1, \ldots, n\}$. Each term inside the sum on the right-hand side of the equation above contains at least one factor equal to ϵ_i for some $i = 1 \ldots, n$. Since the sum contains at most 2^n terms, $n \leq b$, $0 < e^{-z_i k_i/z} \leq 1$, and $\epsilon_i < 2^{-2-10b} < 1$, it follows that

$$E[S_n'|\delta] - \exp\left(-\sum_{i=1}^{n} z_i \delta_i / z\right) < \sum_{k \in \{0,1\}^n} \max_i \epsilon_i < 2^{-2-9b} \,.$$

Consequently, we have that

11

$$qE[S'_n|\delta] - \exp\left(1 - \sum_{i=1}^{n} z_i\delta_i/z\right) =$$

$$\underbrace{qE[S'_n|\delta] - q\exp\left(-\sum_{i=1}^{n} z_i\delta_i/z\right)}_{<q2^{-2-9b}} + \overbrace{[q-e]}^{\leq 2^{-2-9b}} \underbrace{\exp\left(-\sum_{i=1}^{n} z_i\delta_i/z\right)}_{\leq 1},$$

which, since $q < 3$, is strictly smaller than 2^{-9b}. $\qquad\qquad\square$

According to the lemma above, the absolute difference between the expected utility of a strategy δ and the value of the function f evaluated at the point $\sum_i z_i\delta_i/z$ is less than 2^{-9b}. Moreover, $f(\sum_i z_i\delta_i/z)$ is an upper bound on the expected value of any strategy δ. The next result builds on these two facts to show that yes- and no-solutions can be distinguished by the sign of their expected utilities.

Proposition 2. *The partition problem admits a yes-solution if and only if the maximum expected utility of a strategy is nonnegative.*

PROOF. According to Lemma 1, for any strategy δ we have that $E[U|\delta] > f(\sum_{i=1}^{n} z_i\delta_i/z) - 2^{-9b}$. Thus, if a yes-solution exists then $\max_\delta E[U|\delta] > f(1) - 2^{-9b} = 0$. To show that any no-solution has non-positive expected utility, consider the function f in Equation (6), whose graph is the curve in Figure 3 up to the small additive constant 2^{-9b}. Its first and second derivatives are, respectively, $f'(s) = \exp(1-s) - 1$ and $f''(s) = -\exp(1-s)$. Thus, the function is strictly concave, increases for $s < 1$, decreases for $s > 1$, and has a maximum at $s = 1$. This implies that f is maximized by a strategy whose scaled value $s_\delta = \sum_{i=1}^{n} z_i\delta_i/z$ is the closest to one (over all values s_δ induced by strategies). If a yes-solution δ exists, then its scaled value s_δ maximizes f.

Recall that if the partition problem has no yes-solution then any strategy δ satisfies $|z - \sum_i z_i\delta_i| \geq 1$, from which it follows that either $s_\delta = \sum_i z_i\delta_i/z \geq 1 + 1/z$ or $s_\delta = \sum_i z_i\delta_i/z \leq 1 - 1/z$. Thus, the maximum value of f over scaled values of no-solutions is either at $1 - 1/z$ (the scaled value closest to and smaller than one) or at $1 + 1/z$ (the closest to and bigger than one). Since, by Lemma 1, $f(s_\delta)$ is an upper bound on the expected utility of any strategy δ, if a yes-solution does not exist then $\max_\delta E[U|\delta] < \max\{f(1 + 1/z), f(1 - 1/z)\}$. Consider the difference $f(1+1/z) - f(1-1/z) = -\exp(-1/z) + \exp(1/z) - 2/z$. By analyzing its first and second derivatives, one can show that the difference is a strictly convex function of z whose infimum is zero. Hence, the difference is positive (i.e., $f(1+1/z) > f(1-1/z)$), and it suffices for the result to show that $f(1+1/z)$ is negative for any positive integer z. For any instance of the partition problem, we have that $2z \leq 2^b$, from which it follows that $2^{-9b} < z^{-3}/8$. By definition, $f(1+1/z) = 1 - e^{-1/z} - z^{-1} + 2^{-9b} < 1 - e^{-1/z} - z^{-1} - z^{-3}/8$, which according to Lemma 5 in the appendix is negative for any instance. $\qquad\square$

We finally get to the desired hardness result.

Theorem 3. *Given a polytree-shaped limid of bounded treewidth over binary variables, deciding whether there is a strategy whose expected utility exceeds a given rational number is NP-complete.*

PROOF. Since the diagram is polytree-shaped, we can compute the expected utility of any strategy in polynomial time by e.g. variable elimination, which shows membership in NP. According to Proposition 2, for any instance of the partition problem we can build in polynomial time a polytree-shaped influence diagram of bounded treewidth with only binary variables and such that the partition problem has a yes-solution if and only the optimum strategy has non-negative expected utility. Hence, deciding whether the maximum expected utility of a strategy exceeds any given threshold solves the partition problem, and is thus NP-hard. □

5. Conclusion

Finding an optimal strategy for limited memory influence diagrams is known to be NP-hard even for polytree-shaped diagrams with ternary variables and a single value node [7]. When the diagram has a single value node, the problem can be mapped into one of computing marginal probability bounds in a properly designed credal polytree [12]. We showed here that this correspondence between influence diagrams and credal networks allow us to solve polytree-shaped diagrams with binary variables and a single value node in polynomial time, as this is the case for computing marginal probability bounds in credal polytrees [17]. The mapping however does not work on diagrams that contain multiple value nodes (even if variables are binary), and the theoretical complexity of the strategy selection problem in polytree-shaped diagrams with binary variables and multiple value nodes was until now unknown.

In this paper, we showed by a reduction of the partition problem that selecting optimal strategies in polytree-shaped influence diagrams with binary variables and multiple value nodes is NP-hard. This result closes the fixed-parameter complexity analysis of planning with limited memory influence diagrams in what concerns the topology of the underlying graph and the cardinality of the variables. In summary, selecting optimal strategies in polytree-shaped limids with binary variables and a single value node is polynomial-time solvable, and relaxing any of these conditions leads to NP-hard problems.

Acknowledgments

References

[1] S. L. Lauritzen, D. Nilsson, Representing and solving decision problems with limited information, Management Science 47 (2001) 1235–1251.

[2] E. A. Hansen, Solving POMDPs by searching in policy space, in: Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence (UAI), 1998, pp. 211–219.

[3] D. S. Bernstein, E. A. Hansen, S. Zilberstein, Bounded policy iteration for decentralized POMDPs, in: Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence (IJCAI), 2005, pp. 1287–1292.

[4] A. Detwarasiti, R. D. Shachter, Influence diagrams for team decision analysis, Decision Analysis 2 (4) (2005) 207–228.

[5] C. Amato, D. S. Bernstein, S. Zilberstein, Optimizing fixed-size stochastic controllers for POMDPs and decentralized POMDPs, Autonomous Agents and Multi-Agent Systems 21 (3) (2010) 293–320.

[6] X. Wu, A. Kumar, S. Zilberstein, Influence diagrams with memory states: Representation and algorithms, in: Proceedings of the Second International Conference on Algorithmic Decision Theory, 2011, pp. 306–319.

[7] D. D. Mauá, C. P. de Campos, M. Zaffalon, Solving limited memory influence diagrams, Journal of Artificial Intelligence Research 44 (2012) 97–140.

[8] D. D. Mauá, C. P. de Campos, M. Zaffalon, The complexity of approximately solving influence diagrams, in: Proceedings of the 28th Conference on Uncertainty in Artificial Intelligence (UAI), 2012, pp. 604–613.

[9] D. D. Mauá, C. P. de Campos, Solving decision problems with limited information, in: Advances in Neural Information Processing Systems 24 (NIPS), 2011, pp. 603–611.

[10] A. Antonucci, S. Yi, C. P. de Campos, M. Zaffalon, Generalized loopy 2U: a new algorithm for approximate inference in credal networks, International Journal of Approximate Reasoning 55 (5) (2010) 474–484.

[11] Q. Liu, A. Ihler, Belief propagation for structured decision making, in: Uncertainty in Artificial Intelligence (UAI), AUAI Press, 2012, pp. 523–532.

[12] C. P. de Campos, Q. Ji, Strategy selection in influence diagrams using imprecise probabilities, in: Proceedings of the 24th Conference in Uncertainty in Artificial Intelligence (UAI), 2008, pp. 121–128.

[13] C. Yuan, X. Wu, E. A. Hansen, Solving multistage influence diagrams using branch-and-bound search, in: Proceedings of the 26th Conference on Uncertainty in Artificial Intelligence (UAI), 2010, pp. 691–700.

[14] A. Antonucci, M. Zaffalon, Decision-theoretic specification of credal networks: a unified language for uncertain modeling with sets of Bayesian networks, International Journal of Approximate Reasoning 49 (2) (2008) 345–361.

[15] F. G. Cozman, Credal networks, Artificial Intelligence 120 (2000) 199–233.

[16] C. P. de Campos, F. G. Cozman, The inferential complexity of Bayesian and credal networks, in: Proceedings of the 19th International Joint Conference on Artificial Intelligence (UAI), 2005, pp. 1313–1318.

[17] E. Fagiuoli, M. Zaffalon, 2U: An exact interval propagation algorithm for polytrees with binary variables, Artificial Intelligence 106 (1) (1998) 77–107.

[18] R. A. Howard, J. E. Matheson, Influence diagrams, in: Readings on the Principles and Applications of Decision Analysis, Strategic Decisions Group, 1984, pp. 721–762.

[19] J. A. Tatman, R. D. Shachter, Dynamic programming and influence diagrams, IEEE Transactions on Systems, Man and Cybernetics 20 (2) (1990) 365–379.

[20] F. V. Jensen, T. D. Nielsen, Bayesian Networks and Decision Graphs, 2nd Edition, Information Science and Statistics, Springer, 2007.

[21] E. Fagiuoli, M. Zaffalon, A note about redundancy in influence diagrams, International Journal of Approximate Reasoning 19 (3-4) (1998) 351–365.

[22] R. D. Shachter, Bayes-ball: The rational pastime (for determining irrelevance and requisite infomation in belief networks and influence diagrams), in: Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence (UAI), 1998, pp. 480–487.

[23] T. D. Nielsen, F. V. Jensen, Well-defined decision scenarios, in: Proceedings of the 15th Conference on Uncertainty in Artificial Intelligence (UAI), 1999, pp. 502–511.

[24] R. D. Shachter, Efficient value of information computation, in: Proceedings of the 15th Conference on Uncertainty in Artificial Intelligence (UAI), 1999, pp. 594–601.

[25] G. F. Cooper, A method for using belief networks as influence diagrams, *Fourth Workshop on Uncertainty in Artificial Intelligence* (1988).

[26] M. R. Garey, D. S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, W. H. Freeman, 1979.

Appendix A. Supplementary Results

The following lemma is used to prove Lemma 1.

Lemma 4. *Let $0 < x \leq 2$ be a real number, $M \geq 12$ be an even integer, and $r = \sum_{k=0}^{M}(-1)^k x^k/k!$. It follows that $e^{-x} < r < e^{-x} + 2^{-M}$.*

PROOF. According to Taylor's theorem, the number e^{-x} can be approximated by M-th order Taylor expansion of the exponential function around zero, that is,

$$e^{-x} = \sum_{k=0}^{M} (-1)^k \frac{x^k}{k!} + R_M(x),$$

where the term $R_M(x)$ denotes the approximation error and is called the residual. Note that the first term on the right is by definition the number r. The residual $R_M(x)$ is well-known to satisfy

$$R_M(x) = (-1)^{M+1} e^{-\xi} \frac{x^{M+1}}{(M+1)!},$$

for some $0 < \xi < x$. Since M is even, the factor $(-1)^{M+1} = -1$ and $R_M(x) < 0$, whence $r > \exp(-x)$. Using the inequality $e^{-\xi} < 1$ (valid for any $\xi > 0$), we find that

$$|R_M(x)| < \frac{x^{M+1}}{(M+1)!}.$$

Since $x \leq 2$, and $(k/3)^k < k!$ (for any positive integer k), the right-hand side of the inequality above satisfies

$$\frac{x^{M+1}}{(M+1)!} < \frac{2^{M+1}}{([M+1]/3)^{M+1}} = \frac{2^{M+1} 3^{M+1}}{(M+1)^{M+1}},$$

and because $M \geq 12 = 2^2 \cdot 3$, we have that

$$\frac{2^{M+1} 3^{M+1}}{(M+1)^{M+1}} < \frac{2^{M+1} 3^{M+1}}{2^{2(M+1)} 3^{M+1}} = \frac{1}{2^{M+1}} < \frac{1}{2^M}.$$

The result follows from $r = e^{-x} - R_M(x)$. □

The following result is used to prove Proposition 2.

Lemma 5. *For any positive integer z and real $r < 1/3$, it follows that*

$$1 - e^{-1/z} - \frac{1}{z} + \frac{r}{z^3} < 0.$$

PROOF. Consider the second-order Taylor expansion of e^{-x} around zero given by $T_2(x) \stackrel{\text{def}}{=} 1 - x + x^2/2$, whose residual satisfies

$$R_2(x) \stackrel{\text{def}}{=} e^{-x} - T_2(x) = -x^3 e^{-\xi}/6 > -x^3/6,$$

for some number ξ between 0 and x. Hence, $-e^{-x} < x^3/6 - T_2(x)$, from which it follows that

$$1 - e^{-1/z} - \frac{1}{z} + \frac{r}{z^3} < 1 + \left[\frac{1}{6z^3} - T_2(1/z) \right] - \frac{1}{z} + \frac{r}{z^3},$$

which equals $r/z^3 - 1/(2z^2) + 1/(6z^3) = (2r - z + 1/3)/(2z^3) < 0$. □