Ann Math Artif Intell (2017) 81:273–313
DOI 10.1007/s10472-017-9553-y

A symbolic algebra for the computation of expected utilities in multiplicative influence diagrams

Manuele Leonelli[1] · **Eva Riccomagno**[2] · **Jim Q. Smith**[3]

Published online: 21 June 2017

Abstract Influence diagrams provide a compact graphical representation of decision problems. Several algorithms for the quick computation of their associated expected utilities are available in the literature. However, often they rely on a full quantification of both probabilistic uncertainties and utility values. For problems where all random variables and decision spaces are finite and discrete, here we develop a symbolic way to calculate the expected utilities of influence diagrams that does not require a full numerical representation. Within this approach expected utilities correspond to families of polynomials. After characterizing their polynomial structure, we develop an efficient symbolic algorithm for the propagation of expected utilities through the diagram and provide an implementation of this algorithm using a computer algebra system. We then characterize many of the standard manipulations of influence diagrams as transformations of polynomials. We also generalize the decision analytic framework of these diagrams by defining asymmetries as operations over the expected utility polynomials.

Keywords Asymmetric decision problems · Computer algebra · Influence diagrams · Symbolic inference

Mathematics Subject Classification (2010) 68T37

Manuele Leonelli was funded by Capes, whilst J.Q. Smith was partly supported by EPSRC grant EP/K039628/1 and The Alan Turing Institute under EPSRC grant EP/N510129/1.

✉ Manuele Leonelli
manuele.leonelli@glasgow.ac.uk

[1] School of Mathematics and Statistics, University of Glasgow, Glasgow G12 8QW, UK

[2] Dipartimento di Matematica, Universita' degli Studi di Genova, Via Dodecaneso 35, 16146 Genova, Italia

[3] Department of Statistics, The University of Warwick, Coventry CV47AL, UK

1 Introduction

Decision makers (DMs) are often required to choose in critical situations between a wide range of different alternatives. They need to consider the mutual influence of quantifications of different types of uncertainties, the relative values of competing objectives together with the consequences of the decisions they will make. They can thus benefit from an intuitive framework which draws together these uncertainties and values so as to better understand and evaluate the full consequences of the assumptions they are making. To this end, a variety of graphical models have been developed. The most important of these are *Bayesian networks* (BNs) [37, 44] and *influence diagrams* (IDs) [4, 26, 36], both of which provide an intuitive qualitative representation of the elements of the DM's problem together with relatively fast computational tools for the calculation of, respectively, probabilities and expected utilities (EUs) [27, 39, 44]. Although only the second class of models can be used to automatically select an optimal course of action, i.e. an EU maximizing decision, both BNs and IDs are invaluable decision support tools, enabling DMs to easily investigate the effect of their inputs to an output of interest.

Most of the algorithms for the computation of probabilities and EUs rely on a full specification of the model's parameters. Furthermore, commonly available software almost exclusively work numerically with complete elicitations. However, often in practice DMs might not be confident about the precision of their specifications, nor have available all such values. This may lead to non-robust decision making where the efficacy of decisions can change under small perturbations of the model's inputs. Symbolic approaches, not requiring full elicitations of the parameters, have proven useful in performing these types of input-output investigations, usually called *sensitivity analyses*, both in fully inferential and decision making contexts [1, 2, 14, 35]. A variety of symbolic methods for both inference and sensitivity analysis are now in place for BNs [10, 12]. However, the development of symbolic techniques for EU computations in IDs has been largely neglected. An exception is a recent paper [6] where *decision network polynomials* are defined in the context of Bayesian decision problems. These are piece-wise functions made of so-called *pieces*: multilinear polynomials having as indeterminates both probability and utility parameters. A new symbolic sensitivity technique is then developed in [6] based on differentiation and difference operators.

In this paper, we focus on a large class of IDs called *multiplicative influence diagrams* (MIDs), which include as a special case standard IDs equipped with additive utility factorizations, and fully characterize the polynomial structure of the EU pieces (Section 3). We then introduce a symbolic algorithm for their computation, based on simple matrix operations (Section 4), and its implementation in the computer algebra system MapleTH[1] (Appendix B). Because of the simplicity of the required operations, our algorithm is shown to have computational times comparable to those of standard numerical evaluation software for graphical models (Section 4.4). In contrast to standard software, which assumes an additive factorization between utility nodes, we also explicitly analyze cases when the more general class of multiplicative utility functions might be necessary [29, 30, 44]. We concentrate our study on the class of multiplicative factorizations because this provides some computational advantages over, for example, the more general class of multilinear utilities [30], whilst allowing for enough flexibility to model the DM's preferences in many real

[1]Maple is a trademark of Waterloo Maple Inc.

cases [22, 29]. This factorization turns out to be particularly efficient since it leads to a distributed propagation of EUs as shown in Proposition 1.

The symbolic definition of the ID's probabilities and utilities in Section 3 provides an elegant and efficient embellishment of the associated graphical representation of the decision problem, around which symbolic computations can then be carried out. In Sections 5 and 6 standard manipulations of IDs and asymmetries are characterised on this new polynomial representation. Importantly we demonstrate that, whilst graphical representations of asymmetries are rather more obscure than standard ID models, in our symbolic approach the imposition of asymmetries greatly simplifies the polynomial representation of the problem. The example in Section 7 then outlines the insights our approach can give to DMs through the comparison of different parameters' specifications. Our symbolic approach has the great advantage in such sensitivity studies that, by exploiting the known polynomial expression of the problem, one can simply plug-in different numerical specifications and instantaneously get the EU values. In standard numerical approaches on the other hand, the evaluation algorithm needs to be run for each combination of parameters considered. This can become quickly unfeasible even for rather small problems.

2 A review of symbolic approaches to decision making and support

Symbolic inference and decision support techniques have already been used for the analysis of BN models. A symbolic definition of probabilities in BNs in terms of multilinear polynomials first appeared in [9]. Since then various inferential techniques have been developed [11, 18, 24]. Their most demonstrably useful application is in the process of validating models through sensitivity analyses. Two main approaches are adopted in practice. The first one is based on differentiation of the probability polynomials and is useful for the analysis of global changes of probability distributions [13, 14]. The second one concerns local changes studied via sensitivity functions [15, 23], which, because of the assumed multilinearity, are simple linear functions of the parameters of interest. Recently, symbolic methods have been extended to asymmetric models [24, 33] where the associated polynomials might not exhibit regular multilinear structures as for BNs.

Although it is known that EUs in IDs also have a multilinear structure [21], symbolic methodologies for such models have not been studied consistently. Only recently the robustness of decision models has been analysed from a symbolic viewpoint in [6]. For the i-th available strategy, [6] defines the functions $u_i : \mathcal{X} \rightarrow \mathbb{R}$, where $\mathcal{X}$ is the parameter space, representing the EU of the associated strategy and called EU piece. The decision network polynomial is then defined as $\max_{i=1,\dots,M} u_i(x)$, for M available strategies, and represents the expected utility of the optimal strategy for the combination of parameters x.

However, no assumptions about the attributes of the problem entertaining various conditional utility or additive/preferential independences are utilized in [6], where a utility value is associated to each possible combination of decisions and realizations of random variables. Such assumptions, often encountered in applied decision analyses, are commonly encoded in a particular factorization of the utility function which then leads to fast and distributed algorithm for the computation of expected utilities. Thus, without formally acknowledging such independences a great amount of information about the preferences of the DM and computational efficiency can be lost. Furthermore, no details on how to compute the functions u_i are given in [6]. In this paper we extend this symbolic framework by developing a distributed symbolic procedure for the computation of the EU pieces for utilities chosen in the large class of multiplicative IDs [30, 44]. We further fully characterize symbolically the functions u_i of the decision problem. This enables the application of the proposed methodology to robustness studies where certain parameters are treated as unknown. In Section 7 via an example we show how to exploit our definition for informing a DM about the optimization process. A full development of such symbolic optimization techniques is beyond the scope of this paper.

Of course the solution and investigation of both generic decision problems and influence diagrams can be performed outside of the full Bayesian symbolic paradigm and using uncertainty calculi that relax the assumption of an exact and complete probability specification. One of such proposals [8], is based on imprecise probabilities and consists of mapping the evaluation of an ID into an inferential problem in credal networks [17], solved using multilinear programming [7]. The objective function of such an optimization problem can be shown to be multilinear and to share many features with our polynomial representation of EUs, although within a different domain. Because of the use of imprecise probabilities the parameters of the decision problem can be specified only partially.

Symbolic evaluation methods have also been introduced for discrete and finite time decision Markov processes that do not require full parameters' elicitations (e.g. [31]). As an ID can always be cast as a Markov decision process, the evaluation methods originally designed for general Markov processes can be straightforwardly applied to IDs. A different approach is taken by the so called symbolic dynamic programming: for such a technique the sample space does not need to be fully specified [38, 47]. Again these methods have the capability of helping the DM to discover the most critical features of the decision problem where accurate specification of inputs is most necessary.

The methods reviewed above propose to automate decision making in a variety of frameworks and reasoning paradigms where DMs do not need to provide complete and/or exact parameters' specifications. These have proven to be successful and computationally efficient, but EU maximization is still most commonly applied within a standard probabilistic domain. Therefore, here we assume that the DM plans to behave as an EU maximizer and we will henceforth work entirely within this most standard framework.

3 Symbolic representation of influence diagrams

In this paper, with the exception of Section 6, we consider those Bayesian decision problems that can be represented by an ID and are usually called *uniform* (or *symmetric*) [32, 44]. Let n be a positive integer ($n \in \mathbb{Z}_{\geq 1}$) and $\mathbb{D}$ and $\mathbb{V}$ be a partition of $[n] = \{1, \dots, n\}$. Let $\{Y_i : i \in \mathbb{D}\}$ be a set of controlled (or decision)[2] variables and $\{Y_i : i \in \mathbb{V}\}$ a set of non-controlled (or random) variables. As in standard ID representations, the set of decision variables is assumed to be totally ordered and the union of $\{Y_i : i \in \mathbb{V}\}$ and $\{Y_i : i \in \mathbb{D}\}$ to be totally ordered compatibly with a partial order on the random variables. Let $\preceq$ be the chosen ordering relationship. The ordering on the Y_i's is reflected by their indices, that is if $Y_i \preceq Y_j$ then $i < j$.

For $i \in [n]$ and $r_i \in \mathbb{Z}_{\geq 1}$, let $[r_i]_0 = \{0, \dots, r_i - 1\}$ and Y_i take values in $\mathcal{Y}_i = [r_i]_0$. For $A \subseteq [n]$, let the vector $Y_A = (Y_i)_{i \in A}$ take values in $\mathcal{Y}_A = \times_{i \in A} \mathcal{Y}_i$ and denote with y_A a generic instantiations of Y_A. Examples of this notation are: the vector $Y_{[n]}$ includes all the variables, whilst $Y_{\mathbb{D}}$ and $Y_{\mathbb{V}}$ are the vectors of controlled and random variables respectively.

[2]With controlled variable we mean a variable set by the DM to take a particular value.

3.1 Multiplicative influence diagrams

We consider the class of *multiplicative* IDs entertaining a multiplicative factorization over the utility nodes $U = (U_1, \ldots, U_m)^\mathrm{T}$ (see e.g. [30, 44]). For $i \in [m]$, U_i is a function onto $[0, 1]$ defined on a subspace $\mathcal{Y}_{P_i}$ of $\mathcal{Y}_{[n]}$ where $P_i \subseteq [n]$ is assumed non empty.

Definition 1 A **multiplicative influence diagram** (MID) G consists of three components: a directed acyclic graph (DAG) with vertex (or node) set $V(G) = Y_{[n]} \cup U$, a transition probability function related to the random variables $Y_{\mathbb{V}}$ and a multiplicative factorization function related to the U nodes.

Example 1 Figure 1 presents an MID with $n = 6$, $m = 3$, $\mathbb{D} = \{1, 4\}$, $\mathbb{V} = \{2, 3, 5, 6\}$ and vertex set $V(G) = \{Y_1, \ldots, Y_6, U_1, \ldots, U_3\}$. There are two controlled variables, Y_1 and Y_4, four random variables, Y_2, Y_3, Y_5 and Y_6, and three utility nodes, U_1, U_2 and U_3. We adopt the convention by which decision variables and random variables are respectively framed with squares and circles. All variables are binary and take values in the spaces $\mathcal{Y}_i = \{0, 1\}$, $i \in [6]$.

Next we describe the three components of an MID starting from its edge (or arc) set $E(G)$. For $i \in [n]$, the parent set of Y_i is the sub-vector of $Y_{[n]}$ indexed by $\Pi_i \subset [i-1]$. For $i \in [m]$, the parent set of U_i is the sub-vector Y_{P_i} of $Y_{[n]}$ where $P_i \subseteq [n]$ is the non empty set mentioned above and thus each utility node has at least one parent. Furthermore any two P_i's are assumed disjoint so that each component of $Y_{[n]}$ is parent of at most one utility node. There are three types of edges in an MID:

1. those into U vertices: for $i \in [m]$, U_i has no children and its parent set Y_{P_i} is described above;
2. those into $\mathbb{D}$ vertices: for $i \in \mathbb{D}$, the parent set of Y_i consists of the variables, controlled and non-controlled that are known when Y_i is controlled;
3. those into $\mathbb{V}$ vertices: for $i \in \mathbb{V}$, the parent set of Y_i is such that Y_i is conditionally independent (with respect to the probability law in Definition 1) of the random variables preceding it given its parents and for all instantiations of decisions preceding Y_i.

Recalling that $\Pi_i \subset [i-1]$, Item (3) above can be formulated as $Y_i \perp\!\!\!\perp Y_{[i-1]} \mid Y_{\Pi_i}$, where $\perp\!\!\!\perp$ denotes the *extended* conditional independence operator [19]. This means that standard conditional independence, namely $Y_i \perp\!\!\!\perp Y_{[i-1]\cap\mathbb{V}} \mid Y_{\Pi_i\cap\mathbb{V}}$, holds for all instantiations of the decision variables $Y_{[i-1]\cap\mathbb{D}}$ preceding Y_i. The transition probability function for the random vector $Y_{\mathbb{V}}$ in Definition 1 is given in terms of probability density as the product of

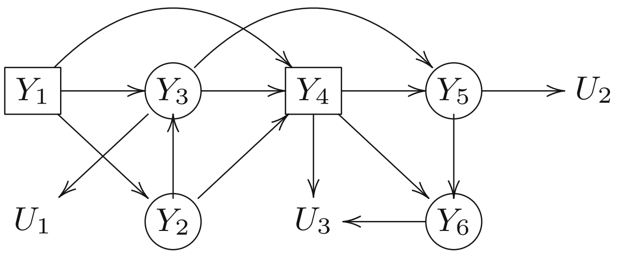


Fig. 1 An MID consisting of *two* decision nodes, Y_1 and Y_4, *four* random nodes, Y_2, Y_3, Y_5 and Y_6, and *three* utility nodes, U_1, U_2 and U_3

$P_i(y_i \mid y_{\Pi_i}) = P(Y_i = y_i \mid Y_{\Pi_i} = y_{\Pi_i})$ for $i \in \mathbb{V}$. Note that y_{Π_i} includes instantiations of controlled variables as well as random variables.

Example 2 The edge set of the MID in Fig. 1 is such that no variable is observed before controlling Y_1, whilst Y_1, Y_2 and Y_3 are observed before controlling Y_4 since $\Pi_4 = \{1, 2, 3\}$. Furthermore its DAG implies that $Y_5 \perp\!\!\!\perp Y_1, Y_2 \mid Y_3, Y_4$ and $Y_6 \perp\!\!\!\perp Y_1, Y_2, Y_3 \mid Y_4, Y_5$. The parent sets of the utility nodes are $P_1 = \{3\}$, $P_2 = \{5\}$ and $P_3 = \{4, 6\}$.

The third component of an MID is a utility function U defined over $\mathcal{Y}_{[n]}$ as

$$U(y_{[n]}) = \begin{cases} \sum_{i \in [m]} k_i U_i(y_{P_i}), & \text{if } h = 0, \\ \sum_{I \in \mathcal{P}_0([m])} h^{n_I - 1} \prod_{i \in I} k_i U_i(y_{P_i}), & \text{otherwise.} \end{cases} \tag{1}$$

where $k_i \in (0, 1)$ is a *criterion weight* [30]; as mentioned above U_i is a function of the random and decision variables in Y_{P_i}. It gives the contribution to the utility function of the controlled and random variables in Y_{P_i} and it does so linearly if $h = 0$, i.e. the first case of (1). It is worthwhile recalling that the Y_{P_i}'s are disjoint. In the second case of (1) h is the unique non-zero solution not smaller than minus one to

$$1 + h = \prod_{i \in [m]} (1 + hk_i). \tag{2}$$

and $\mathcal{P}_0(\cdot)$ denotes the power set without the empty set, n_I is the number of elements in the set I. For $h = 0$, the multiplicative factorisation of an MID, $U(y_{[n]})$, is a weighted sum of the terms $U(y_{P_i})$: thus coinciding with the class of commonly used additive factorizations [29]. Therefore the methodology we develop here applies to utility factorizations of additive form, or *additive IDs*, as well. For $h \neq 0$ the function $U(y_{[n]})$ is a linear combination of all square free products of the U_i's (excluding 1). The h balances the weight of the interaction terms: the larger h is, the bigger is the impact of high order terms.

Example 3 The multiplicative utility factorization associated to the MID in Fig. 1, for $h \neq 0$ and leaving the functions' arguments implicit, can be written as

$$U = k_1U_1 + k_2U_2 + k_3U_3 + hk_1k_2U_1U_2 + hk_1k_3U_1U_3 + hk_2k_3U_2U_3 + h^2k_1k_2k_3U_1U_2U_3.$$

This expression emphasizes the generality of multiplicative utilities, since an additive utility is obtained by setting $h = 0$ and is the sum of the first three terms.

Item 1 above, describing the edges into the utility nodes, extends the total order over $Y_{[n]}$ to $V(G)$. Indeed for $i, j \in [m]$, U_i succeeds U_j and $i > j$ if there exists a parent of U_i which succeeds all parents of U_j in the order $\preceq$ over $Y_{[n]}$: formally, if there is a $k \in P_i$ such that for every $l \in P_j$, $k > l$. For $i \in [m]$, let j_i be the highest index of P_i and $\mathbb{J} = \{j_1, \dots, j_m\}$. The set $\mathbb{J}$ of the greatest parents of the utility nodes in $\preceq$ is fundamental for the Algorithm 4.2 in Section 4.3 because it allows for the computation of the least number of expected utilities by processing a U_i in the algorithm only when strictly necessary. The MapleTH function `CompJ` in Appendix B.1 computes the set $\mathbb{J}$ for a given MID. The totally ordered sequence of $V(G)$ is called **decision sequence (DS)** of the MID G and is denoted by $S := (Y_1, \dots, Y_{j_1}, U_1, Y_{j_1+1}, \dots, Y_{j_m}, U_m)$. As in [3], we do not introduce utility nodes only at the end of the DS. This enables us to base the choice of optimal decisions, through the algorithm given below, only on the values of the relevant attributes.

Example 4 The DS of the MID in Fig. 1 is $(Y_1, Y_2, Y_3, U_1, Y_4, Y_5, U_2, Y_6, U_3)$ with $j_1 = 3$, $j_2 = 5$, $j_3 = 6$ and thus $\mathbb{J} = \{3, 5, 6\}$.

3.2 Evaluation of MIDs

In this section we set the background for an efficient symbolic algorithm for *evaluating* an MID, namely for computing the expected value of (1) for all possibile decisions $y_{\mathbb{D}} \in Y_{\mathbb{D}}$ and identifying a sequence of optimal decisions that maximizes it. We do this by exploiting the sequential structure of (1) which by linearity is transferred to its EU function. However, this can be done only for MIDs in *extensive form* [42], namely those MIDs whose topology is such that, for any index $j \in \mathbb{D}$, only variables that are known at the time the DM makes the decision Y_j have an index lower than j. This is because the evaluation will output optimal decisions as functions of observed quantities only [44]. Extensive form is thus a property referring to the edges into the decision variables of an MID.

Definition 2 An MID G is said to be in **extensive form** if Y_i is a parent of Y_j, $j \in \mathbb{D}$, for all $i < j$.

Example 5 The MID in Fig. 1 is in extensive form since $\Pi_4 = \{1, 2, 3\}$. If either the edge (Y_2, Y_4) or (Y_3, Y_4) were deleted then the MID would not be in extensive form.

We first study MIDs in extensive form and only in Section 5 we consider manipulations of non extensive MIDs which turn them into extensive form. Without loss of generality we assume that any vertex corresponding to a variable in $Y_{[n]}$ has at least one child. Indeed, random and controlled vertices with no children could simply be deleted from the graph without changing the outcome of the evaluation [32]. In Example 5 the only vertices with no children are utility nodes.

A typical way to evaluate an MID in extensive form is through a backward inductive algorithm on the vertices of the DAG. We present a computationally efficient version of this algorithm, which at each step only utilises the strictly necessary utility nodes. The identification of the optimal policy is based on the computation of the functions $\bar{U}_i(y_{B_i})$, $i \in [n]$, which are formally introduced in Proposition 1 and each of which depends only on the variables in $Y_{[n]}$ that are strictly required for an MID evaluation. For $i \in [n]$, the set

$$B_i = \left\{ \bigcup_{\substack{k \geq i \\ k \in \mathbb{V}}} \Pi_k \bigcup \bigcup_{\substack{j \geq i \\ j \in \mathbb{J}}} P_j \right\} \setminus \{i, \dots, n\},$$

defines the index sets of the subset of $Y_{[n]}$ which appear as arguments of $\bar{U}_i$. The function `CompBi` in Appendix B.1 computes the B_i's given the definition of an MID. Specifically a set B_i includes only indices smaller than i that are either in the parent set of a random variable Y_k, $k > i$, following Y_i in the DAG or in a set P_j such that U_j succeeds Y_i in the DS of the MID.

Example 6 For the MID in Fig. 1 the set $B_5 = \{3, 4\}$ since $B_5 = \{\Pi_6 \cup \Pi_5 \cup P_3 \cup P_2\} \setminus \{5, 6\}$, $\Pi_6 = \{4, 5\}$, $\Pi_5 = \{3, 4\}$, $P_3 = \{4, 6\}$ $P_2 = \{5\}$, whilst $B_4 = \{3\}$ since $B_4 = \{\Pi_5 \cup \Pi_6 \cup P_2 \cup P_3\} \setminus \{4, 5, 6\} = B_5 \setminus \{4\}$.

Proposition 1 *The optimal decision associated to an MID yields EU equal to $\bar{U}_1(y_{B_1})$ obtained with a backward recursion as follows. For $i \in [n]$ the function $\bar{U}_i(y_{B_i})$ is defined according to whether Y_i is a decision or a random variable as*

$$\bar{U}_i(y_{B_i}) = \begin{cases} \bar{U}_{i,\mathbb{D}}(y_{B_i}), & \text{if } i \in \mathbb{D}, \\ \bar{U}_{i,\mathbb{V}}(y_{B_i}), & \text{if } i \in \mathbb{V} \end{cases}$$

and three cases are distinguished

1. *for $i = n$ either*

$$\begin{aligned} \bar{U}_{n,\mathbb{D}}(y_{B_n}) &= \max_{\mathcal{Y}_n} k_m U_m(y_{P_m}) \quad \text{or} \\ \bar{U}_{n,\mathbb{V}}(y_{B_n}) &= \sum_{y_n \in \mathcal{Y}_n} k_m U_m(y_{P_m}) P_n(y_n \mid y_{\Pi_n}) \end{aligned} \tag{3}$$

2. *for $i \in [n-1]$, $i \in \mathbb{J}$ and $i \in P_l$, then either*

$$\begin{aligned} \bar{U}_{i,\mathbb{D}}(y_{B_i}) &= \max_{\mathcal{Y}_i} \Big(hk_l U_l(y_{P_l}) \bar{U}_{i+1}(y_{B_{i+1}}) + k_l U_l(y_{P_l}) + \bar{U}_{i+1}(y_{B_{i+1}}) \Big) \quad \text{or} \\ \bar{U}_{i,\mathbb{V}}(y_{B_i}) &= \sum_{y_i \in \mathcal{Y}_i} \Big(hk_l U_l(y_{P_l}) \bar{U}_{i+1}(y_{B_{i+1}}) + k_l U_l(y_{P_l}) \\ &\qquad + \bar{U}_{i+1}(y_{B_{i+1}}) \Big) P_i(y_i \mid y_{\Pi_i}), \end{aligned} \tag{4}$$

3. *for $i \in [n-1]$ and $i \notin \mathbb{J}$ either*

$$\begin{aligned} \bar{U}_{i,\mathbb{D}}(y_{B_i}) &= \max_{\mathcal{Y}_i} \bar{U}_{i+1}(y_{B_{i+1}}) \quad \text{or} \\ \bar{U}_{i,\mathbb{V}}(y_{B_i}) &= \sum_{y_i \in \mathcal{Y}_i} \bar{U}_{i+1}(y_{B_{i+1}}) P_i(y_i \mid y_{\Pi_i}). \end{aligned} \tag{5}$$

All maxima and summations in Proposition 1 are over one $\mathcal{Y}_i$ sample space only. For example (3) consists of either a marginalization or a maximization over $\mathcal{Y}_n$ since Y_n is a parent of U_m by construction. The proof of Proposition 1 is in Appendix A.1. Since the algorithm in Proposition 1 consists of a backward inductive routine, its complexity is at best $O(n\exp(t))$ as in standard dynamic programming evaluation of influence diagrams [46], where n is the number of vertices and t is the so called treewidth of the ID [32].

Example 7 To illustrate Proposition 1, we follow the algorithm for the first three steps of the evaluation of the MID in Fig. 1. Since the variable with the highest index, Y_6, is random, the backward induction procedure in Proposition 1 starts using the summation case of (3), specifically

$$\bar{U}_6(y_{B_6}) = \bar{U}_{6,\mathbb{V}}(y_4, y_5) = \sum_{y_6 \in \mathcal{Y}_6} k_3 U_3(y_4, y_6) P(y_6 \mid y_4, y_5).$$

Next the algorithm considers another random variable, Y_5. Since 5 is the highest (and only) index in P_2, the backward induction is based on the summation in (4), which in this case equals

$$\bar{U}_5(y_{B_5}) = \sum_{y_5 \in \mathcal{Y}_5} \big(hk_2 U_2(y_5) \bar{U}_6(y_{B_6}) + k_2 U_2(y_5) + \bar{U}_6(y_{B_6}) \big) P(y_5 \mid y_3, y_4).$$

The backward induction has now reached Y_4, the first decision node. Although Y_4 is an argument of a utility function, it is not the highest index in P_3 and thus the algorithm uses (5) as

$$\bar{U}_4(y_{B_4}) = \bar{U}_{4,\mathbb{D}}(y_3) = \max_{y_4 \in \mathcal{Y}_4} \bar{U}_5(y_{B_5}).$$

We now arrange the EUs, that describe the effectiveness of the available decisions, in a vector as follows.

Definition 3 We define the **EU vector** $\bar{U}_i$, $i \in [n]$, as

$$\bar{U}_i = (\bar{U}_i(y_{B_i}))^{\mathrm{T}}_{y_{B_i} \in \mathcal{Y}_{B_i}}. \tag{6}$$

3.3 Polynomial structure of expected utility

Generalizing work in [9, 18], we introduce a symbolic representation of both the probabilities and the utilities of an MID. For $i \in \mathbb{V}$, $j \in [m]$, $y \in \mathcal{Y}_i$, $\pi \in \mathcal{Y}_{\Pi_i}$ and $\sigma \in \mathcal{Y}_{P_j}$, we define the parameters

$$p_{iy\pi} = P(Y_i = y \mid Y_{\Pi_i} = \pi) \qquad \text{and} \qquad \psi_{j\sigma} = U_j(\sigma).$$

The first index of $p_{iy\pi}$ and $\psi_{j\sigma}$ refers to the random variable and utility vertex to which the parameter is related, respectively. The second index of $p_{iy\pi}$ relates to the state of the random variable, whilst the third one to the parents' instantiation. The second index of $\psi_{j\sigma}$ corresponds to the instantiation of the arguments of the utility function U_j. We take the indices within π and σ to be ordered from left to right in decreasing order, so that e.g. p_{6101} for the diagram of Fig. 1 corresponds to $P(Y_6 = 1 \mid Y_5 = 0, Y_4 = 1)$. The *probability* and *utility vectors* are given by $p_i = (p_{iy\pi})^{\mathrm{T}}_{y \in \mathcal{Y}_i, \pi \in \mathcal{Y}_{\Pi_i}}$ and $\psi_j = (\psi_{j\pi})^{\mathrm{T}}_{\pi \in \mathcal{Y}_{P_j}}$, respectively. Parameters are listed within p_i and ψ_j according to a reverse lexicographic order over their indices [16].[3] In contrast to [6], we use different symbols for utilities and probabilities. This is not only because these are formally different, but also because sensitivity methods can be tailored for these two types of indeterminates separately [35].

Example 8 The symbolic parametrization of the MID in Fig. 1 is summarized in Table 1. This is completed by the definition of the criterion weights k_i and h as in (1)–(2). In Appendix B.5 we report the symbolic definition of this MID using our MapleTH code.

Because probabilities sum to one, for each i and π one of the parameters $p_{iy\pi}$ can be written as one minus the sum of the others. Another constraint is induced by (2) on the criterion weights. However, unless otherwise indicated, we take all the parameters to be unconstrained. Any unmodelled constraint can be added subsequently when investigating the geometric features of the *admissible domains* [35], i.e. regions of the parameters' space over which the preferred strategy does not change.

In the above parametrization, $\bar{U}_i$ consists of a vector of polynomials expressed in the unknown quantities $p_{ij\pi}$, $\psi_{j\sigma}$, k_i and h, whose characteristics are specified in Theorem 1.

Theorem 1 *For an MID G and $i \in [n]$, let $c_i = \prod_{j \in B_i} r_j$, U_l be the first utility node following Y_i in the DS of G and, for $l \leq j \leq m$, w_{ij} be the number of random nodes*

[3]Let $\alpha, \beta \in \mathbb{Z}^n$. We say that α precedes β in reverse lexicographic order if the right-most non zero entry of $\alpha - \beta$ is positive.

Table 1 Parameterization associated to the MID in Fig. 1

$p_2 = (p_{211}, p_{201}, p_{210}, p_{200})^{\mathrm{T}}$
$p_3 = (p_{3111}, p_{3011}, p_{3101}, p_{3001}, p_{3110}, p_{3010}, p_{3100}, p_{3000})^{\mathrm{T}}$
$p_5 = (p_{5111}, p_{5011}, p_{5101}, p_{5001}, p_{5110}, p_{5010}, p_{5100}, p_{5000})^{\mathrm{T}}$
$p_6 = (p_{6111}, p_{6011}, p_{6101}, p_{6001}, p_{6110}, p_{6010}, p_{6100}, p_{6000})^{\mathrm{T}}$
$\psi_1 = (\psi_{11}, \psi_{10})^{\mathrm{T}}$, $\psi_2 = (\psi_{21}, \psi_{20})^{\mathrm{T}}$, $\psi_3 = (\psi_{311}, \psi_{301}, \psi_{310}, \psi_{300})^{\mathrm{T}}$

between Y_i and U_j (including Y_i) in the DS of G. Then $\bar{U}_i$ is a vector of dimension c_i whose entries are polynomials including, for $a = l, \dots, m$ and $b = l, \dots, a$, r_{iba} monomials m_{iba} of degree d_{iba}, where

$$r_{iba} = \binom{a-l}{b-l} \prod_{j=i}^{j_a} r_j, \qquad d_{iba} = (b-l) + 2(b-l+1) + w_{ia}, \qquad m_{iba} = h^{b-l} m'_{iba}, \quad (7)$$

with m'_{iba} a square-free monomial of degree $2(b-l+1) + w_{ia}$.

The proof of Theorem 1 is given in Appendix A.2. Equation (7) defines the *structure* of the polynomials $\bar{U}_i$ of the EU. Specifically, a polynomial is specified once its coefficients and its support (i.e. monomials which form the polynomial) are known. By structure of a polynomial we mean the number of monomials in its support and the number of monomials having a certain degree (sum of exponents). An algorithm for computing the polynomials in Theorem 1 is presented in Section 4, whose operations utilise the polynomial structure of EUs. If the MID has one decision node only, then the entries of the EU vector correspond to the pieces defined in [6].

Example 9 For the MID of Fig. 1 the polynomial structure of the entries of $\bar{U}_5$ can be constructed as follows. From $B_5 = \{3, 4\}$ it follows that $c_5 = 4$. Thus, $\bar{U}_5$ is a column vector of dimension 4. From $U_2 \equiv U_l$ it follows that

$$r_{522} = 2,\; r_{523} = 4,\; r_{533} = 4,\; d_{522} = 3,\; d_{523} = 4,\; d_{533} = 7,$$

using the fact that $w_{52} = 1$ and $w_{53} = 2$. All monomials are square-free because the index b of r_{iba} in Theorem 1 is either equal to l or $l+1$. Each entry of $\bar{U}_5$ is a square free polynomial of degree seven consisting of ten monomials: two of degree 3, four of degree 4 and four of degree 7.

Since additive utility factorizations can be seen as special cases of multiplicative ones by setting $h = 0$, it follows that the EU polynomials of an additive ID are square-free.

Corollary 1 *In the notation of Theorem 1, the EU $\bar{U}_i$, $i \in [n]$, of an additive ID G is a vector of dimension c_i whose entries are square free polynomials of degree $w_{im} + 2$ including, for $a = l, \dots, m$, r_{ia} monomials of degree $w_{ia} + 2$, where $r_{ia} = \prod_{j=i}^{j_a} r_j$.*

Proof This follows directly from Theorem 1, since an additive factorization can be derived by setting $n_I - 1$, the exponent of h in (1), equal to zero. This corresponds to fixing $b = l$ in Theorem 1. □

So far we have assumed that the DM has not provided any numerical specification of the uncertainties and the values involved in the decision problem. This occurs for example if the system is defined through sample distributions of data from different experiments, where probabilities are only known with uncertainty. But in practice sometimes the DM is able to elicit the numerical values of some parameters. These numerical values can then simply be substituted to the corresponding probability and utility parameters in the system of polynomials constructed in Theorem 1 employing e.g. a computer algebra system. In such a case the degree of the polynomials and possibly the number of their monomials can decrease dramatically. We present in Section 7 different plausible numerical specifications of the parameters associated with the MID in Fig. 1, and investigate how the outputs of the MID differ for the different quantifications.

4 The symbolic algorithm

In this section we develop an algorithm based on three operations which exploit the polynomial structure of EUs and use only linear algebra calculus. The MapleTH code for their implementation is reported in Appendix B.3.[4] In contrast to other probabilistic symbolic algorithms (e.g. [10]), our procedure sequentially computes only monomials that are part of the EU polynomials and is thus much more efficient.

4.1 A new algebra for MIDs

We need to introduce two procedures entailing a change of dimension of probability, utility and EU vectors, named `EUDuplicationPsi` and `EUDuplicationP`. These are required in order to multiply parameters associated to compatible instantiations only, i.e. if the common conditioning variables associated to the parameters are instantiated to the same value.

Example 10 In Algorithm 4.2 we will need to compute the Schur (or element-wise) product $\circ$ between the probability vector p_6 and the utility vector ψ_3. However, as specified in Table 1, p_6 has length 8, whilst ψ_3 has length 4. This is because Y_5 is a parent of Y_6 but not an argument of U_3. `EUDuplicationPsi` will then be needed to transform ψ_3 to

$$(\psi_{311}, \psi_{301}, \psi_{311}, \psi_{301}, \psi_{310}, \psi_{300}, \psi_{310}, \psi_{300}),$$

so that $p_6 \circ \psi_3$ equals to

$$(\psi_{311}p_{6111}, \psi_{301}p_{6011}, \psi_{311}p_{6101}, \psi_{301}p_{6001},$$
$$\psi_{310}p_{6110}, \psi_{300}p_{6010}, \psi_{310}p_{6100}, \psi_{300}p_{6000}).$$

The above vector then only includes entries associated to compatible instantiations.

For conciseness, we detail here only the `EUDuplicationPsi` procedure and refer to Appendix B.2 for the code of both procedures. The steps of `EUDuplicationPsi` are shown in Algorithm 4.1. For a vector ψ, let $\psi^{s,t}$ be the subvector of ψ including the entries from $s\cdot(t-1)+1$ to $s\cdot t$, for suitable $s, t \in \mathbb{Z}_{\geq 1}$. For $i \in [n-1]$ and $j \in [m]$, the procedure takes 7 elements as input: an EU $\bar{U}_{i+1}$; the utility vector associated to the utility node preceding Y_{i+1}, ψ_j; their dimensions, c_{i+1} and b_j; the sets B_{i+1} and P_j; the dimensions of all the probability vectors of the MID, $r = (r_1, \ldots, r_n)^{\mathrm{T}}$.

[4] Some inputs of the MapleTH functions in Appendix B.3 are different from those used in this section which are chosen to illustrate the procedure as concisely as possible.

Algorithm 4.1 EUDUPLICATIONPSI($\bar{\boldsymbol{U}}_{i+1}, \boldsymbol{\psi}_j, B_{i+1}, P_j, \boldsymbol{r}, c_{i+1}, b_j$)

```
for $k \leftarrow i$ downto 1
    do if $k \in \{\{B_{i+1} \cup P_j\} \setminus \{B_{i+1} \cap P_j\}\}$
        then $w_k = \prod_{l=k+1}^{j} \mathbb{1}_{\{l \in \{B_{i+1} \cup P_j\}\}}(r_l)$
             if $k \in B_{i+1}$
                 then $\boldsymbol{\psi}_j = \left(\underbrace{\boldsymbol{\psi}_j^{w_k,1} \cdots \boldsymbol{\psi}_j^{w_k,1}}_{r_k \text{ times}} \cdots \underbrace{\boldsymbol{\psi}_j^{w_k,c_j/w_k} \cdots \boldsymbol{\psi}_j^{w_k,c_j/w_k}}_{r_k \text{ times}}\right)$
             else if $k \in P_j$
                 then $\bar{\boldsymbol{U}}_{i+1} = \left(\underbrace{\bar{\boldsymbol{U}}_{i+1}^{w_k,1} \cdots \bar{\boldsymbol{U}}_{i+1}^{w_k,1}}_{r_k \text{ times}} \cdots \underbrace{\bar{\boldsymbol{U}}_{i+1}^{w_k,c_i/w_k} \cdots \bar{\boldsymbol{U}}_{i+1}^{w_k,c_i/w_k}}_{r_k \text{ times}}\right)$
return $(\bar{\boldsymbol{U}}_{i+1}, \boldsymbol{\psi}_j)$
```

For all indices smaller than i and not in $B_{i+1} \cap P_j$, Algorithm 4.1 computes a positive integer number w_k equal to the product of the dimension of the probability vectors with index bigger than k belonging to $B_{i+1} \cup P_j$. The index k is either in B_{i+1} or in P_j. When $k \in B_{i+1}$, each block of w_k rows of ψ_j is consecutively duplicated $r_k - 1$ times.

The first of the three operations we introduce is `EUMultiSum`, which computes a weighted multilinear sum between a utility vector and an EU. In the algorithm of Section 4.3, an `EUMultiSum` operation is associated to every utility vertex of the MID. This operation is required to formally assess the impact of a utility vertex to the overall EU and corresponds to a symbolic version of the sums in (4). Let $P = \{P_1, \dots, P_m\}$.

Definition 4 (**EUMultiSum**) For $i \in [n]$, let $\bar{U}_{i+1}$ be an EU vector and ψ_j the utility vector of node U_j, $j \in [m]$, succeeding Y_i in the DS. The `EUMultiSum`, $+^{EU}$, between $\bar{U}_{i+1}$ and ψ_j is defined as

1. $\bar{U}'_{i+1}, \psi'_j \longleftarrow$`EUDuplicationPsi`$(\bar{U}_{i+1}, \psi_j, B_{i+1}, P_j, r, c_{i+1}, b_j)$;
2. $h \cdot k_j \cdot (\bar{U}'_{i+1} \circ \psi'_j) + k_j \cdot \psi'_j + \bar{U}'_{i+1}$, where $\circ$ and $\cdot$ denote respectively the Schur (or element-wise) and the scalar products.

The second operation, `EUMarginalization` is applied to any random vertex of the MID. This operation is the symbolic equivalent of marginalizations (sums) $\sum_{y_i \in \mathcal{Y}_i}$ in Proposition 1, often called variable elimination in the literature [39].

Definition 5 (**EUMarginalization**) For $i \in \mathbb{V}$, let $\bar{U}_{i+1}$ be an EU vector and p_i a probability vector. The `EUMarginalization`, Σ^{EU}, between $\bar{U}_{i+1}$ and p_i is defined as

1. $\bar{U}'_{i+1}, p'_i \longleftarrow$`EUDuplicationP`$(\bar{U}_{i+1}, p_i, \Pi_i, P, r, B_{i+1}, \mathbb{J})$;

2. $I_{i,\mathbb{V}} \times (\bar{U}'_{i+1} \circ p'_i)$, where $\times$ is the standard matrix product and $I_{i,\mathbb{V}}$ is a matrix with $c_{i+1}s_i/r_i \in \mathbb{Z}_{\geq 1}$[5] rows and $c_{i+1}s_i$ columns defined as

$$I_{i,\mathbb{V}} = \big(\big(1\ 0\ \cdots\ 0\big)\ \big(0\ 1\ \cdots\ 0\big)\ \cdots\ \big(0\ 0\ \cdots\ 1\big) \big)^{\mathrm{T}}$$

where 1 and 0 denote row vectors of dimension r_i with all entries equal to one and zero respectively and $s_i = \prod_{k \in \{\Pi_i \setminus B_{i+1}\}} r_k$.

The last operation is a selection of a decision policy $y_i \in \mathcal{Y}_i$ in $\bar{U}_{i+1}$, $i \in \mathbb{D}$, for every element of $\mathcal{Y}_{\Pi(i)}$.

Definition 6 (**EUMaximization**) For $i \in \mathbb{D}$, let $\bar{U}_{i+1}$ be an EU vector. An `EUMaximization` over $\mathcal{Y}_i$, $\max^{EU}_{\mathcal{Y}_i}$, is defined by the following steps:

1. select a $y_i^*(\pi) \in \mathcal{Y}_i$, for $\pi \in \mathcal{Y}_{\Pi(i)}$;
2. $I_{i,\mathbb{D}} \times \bar{U}_{i+1}$, where $I_{i,\mathbb{D}}$ is a matrix with $c_{i+1}/r_i \in \mathbb{Z}_{\geq 1}$ rows and c_{i+1} columns defined as

$$I_{i,\mathbb{D}} = \big(\big(e_{y_i^*(1)}\ 0\ \cdots\ 0\big)\ \big(0\ e_{y_i^*(2)}\ \cdots\ 0\big)\ \cdots\ \big(0\ 0\ \cdots\ e_{y_i^*(c_{i+1}/r_i)}\big) \big)^{\mathrm{T}}$$

where $e_{y_i^*(\pi)}$, $\pi \in [c_{i+1}/r_i]$, is a row vector of dimension r_i whose entries are all zero but the one in position $y_i^*(\pi)$, which is equal to one.

Using the terminology of [2] and [25], `EUMaximization` finds its natural application in *open-loop* analyses, where one policy only is under scrutiny. In this case, the DM can simply fix the decision of interest and `EUMaximization` drops the polynomials associated to non-selected policies.[6] Nevertheless, in *closed-loop* analyses, where policies can vary, and in standard evaluation methods the first item of Definition 6 is critical for `EUMaximization`. It is not within the scope of this paper to present a methodology to identify EU maximizing decisions. However, within our symbolic approach polynomial optimization and semi-algebraic methods can be used to guide the optimization process [5]. In Section 7 we present an example of the insights that the symbolic definition gives during the maximization step of an evaluation.

Since all our operations simply consists of standard and matrix products, the complexity of the algorithm for the symbolic computation of EUs we introduce below can be deduced by establishing the number of multiplications associated to each EU-operation. Formally, an `EUMultiSum` consists of

$$n_i^{\mathrm{sum}} = c_{i+1}s_i(2 + m_{i+1}) + 1$$

multiplications, where m_{i+1} is the number of monomials in each entry of $\bar{U}_{i+1}$, c_{i+1} is the dimension of $\bar{u}_{i+1}$ and s_i is given in Definition 5. An `EUMarginalization` consists of

$$n_i^{\mathrm{marg}} = c_{i+1}s_i m_{i+1} + (c_{i+1}s_i)^2/r_i$$

multiplications (without considering the sparsity of the matrix $I_{i,\mathbb{V}}$), where r_i is the size of the sample space of Y_i. Exploiting the structure of the matrix $I_{i,\mathbb{D}}$, an `EUMaximization` can be coded so that it does not perform any multiplication.

[5]This is so since $c_{i+1} = r_i a_{i+1}$, for an $a_{i+1} \in \mathbb{Z}_{\geq 1}$.

[6]The MapleTH function `EUMaximization` in Appendix B.3 currently calls a subfunction `Maximize`, which randomly picks decisions. However, this can be modified to take into account a fixed policy given as input.

Therefore the `EUMultiSum` and `EUMarginalization` operations have complexity $O(n_i^{\text{sum}})$ and $O(n_i^{\text{marg}})$ respectively.

4.2 Polynomial interpretation of the operations

Each of the above three operations changes the EU vectors and their entries in a specific way we formalize in Proposition 2.

Proposition 2 *For $i \in [n-1]$, let $\bar{U}_{i+1}$ be an EU vector whose entries have the polynomial structure of (7) and let U_j be the vertex preceding Y_{i+1} in the DS. Then in the notation of Theorem 1*

- *$\max_{\mathcal{Y}_i}^{EU} \bar{U}_{i+1}$ has dimension $c_{i+1}/r_i \in \mathbb{Z}_{\geq 1}$ and its entries do not change polynomial structure;*
- *$\bar{U}_{i+1} +^{EU} \psi_j$ has dimension $c_{i+1}t_i$, where $t_i = \prod_{k\in\{P_j\setminus B_{i+1}\}} r_k$, and each of its entries consists of $r_{(i+1)ba}$ monomials of degree $d_{(i+1)ba}$, $r_{(i+1)ba}$ monomials of degree $d_{(i+1)ba}+3$ and one monomial of degree 2;*
- *$\bar{U}_{i+1}\Sigma^{EU} p_i$ has dimension $c_{i+1}s_i/r_i$, where $s_i = \prod_{k\in\{\Pi_i\setminus B_{i+1}\}} r_k$, and each of its entries consists of $r_i r_{(i+1)ba}$ monomials of degree $d_{(i+1)ba}+1$.*

This result directly follows from the definition of the above three operations. An illustration of Proposition 2 is given in Example 11 below.

4.3 An algorithm for the computation of an MID's expected utilities

The algorithm for the computation of an MID's EUs is given in Algorithm 4.2. It receives as input the DS of the MID, S, the sets $\mathbb{J}$, $\mathbb{V}$ and $\mathbb{D}$, and the vectors $p = (p_1, \dots, p_n)^{\mathrm{T}}$, $\psi = (\psi_1, \dots, \psi_m)^{\mathrm{T}}$ and $k = (k_1, \dots, k_m, h)^{\mathrm{T}}$. The algorithm corresponds to a symbolic version of the backward induction procedure working over the elements of the DS explicated in Proposition 1. At each inductive step, a utility vertex is considered together with the variable that precedes it in the DS.

Algorithm 4.2 SYMBOLICEXPECTEDUTILITY($\mathbb{J}, S, \boldsymbol{p}, \boldsymbol{\psi}, \boldsymbol{k}, \mathbb{V}, \mathbb{D}$)

```
$\bar{\boldsymbol{U}}_{n+1} = (0)$                                                    (1)
for $k \leftarrow n$ downto 1                                                         (2)
  do for $l \leftarrow m$ downto 1                                                    (3)
       do if $k = j_l$                                                                (4)
            then if $k \in \mathbb{D}$                                                (5)
                   then $\bar{\boldsymbol{U}}_k = \max_{\mathcal{Y}_k}^{EU} (\bar{\boldsymbol{U}}_{k+1} +^{EU} \boldsymbol{\psi}_l)$     (6)
                   else $\bar{\boldsymbol{U}}_k = \boldsymbol{p}_k\, \Sigma^{EU} (\bar{\boldsymbol{U}}_{k+1} +^{EU} \boldsymbol{\psi}_l)$   (7)
          else if $k \in \mathbb{D}$                                                  (8)
            then $\bar{\boldsymbol{U}}_k = \max_{\mathcal{Y}_k}^{EU} \bar{\boldsymbol{U}}_{k+1}$                          (9)
            else $\bar{\boldsymbol{U}}_k = \boldsymbol{p}_k\, \Sigma_{\mathcal{Y}_k}^{EU} \bar{\boldsymbol{U}}_{k+1}$                   (10)
return $(\bar{\boldsymbol{U}}_1)$                                                     (11)
```

In line (1) the EU $\bar{U}_{n+1}$ is initialized to (0), namely a vector of dimension one including a zero. Lines (2) and (3) index a reverse loop over the indices of the variables and the utility vertices respectively (starting from n and m). If the current index corresponds to a variable preceding a utility vertex in the DS (line 4), then the algorithm jumps to lines (5)-(7). Otherwise it jumps to lines (8)-(10). In the former case, the algorithm computes, depending on whether or not the variable is controlled (line 5), either an `EUMaximization` over $\mathcal{Y}_k$ (line 6) or an `EUMarginalization` (line 7) with p_k, jointly to an `EUMultiSum` with ψ_l. In the other case, `EUMaximization` and `EUMarginalization` operations are performed without `EUMultiSum`. The MapleTH function `SymbolicExpectedUtility` in Appendix B.4 is an implementation of Algorithm 4.2.

Example 11 For the MID in Fig. 1 the `SymbolicExpectedUtility` function first considers the random vertex Y_6 which precedes the utility vertex U_3 and therefore first calls the `EUMultiSum` function. For this MID

$$P_3 = \{4, 6\},\ t_6 = 4,\ \Pi_6 = \{4, 5\},\ s_6 = 2.$$

Thus, first $\bar{U}_7$ is replicated four times (since $t_6 = 4$) via `EUDuplicationPsi` and

$$\bar{U}_7 +^{EU} \psi_3 = \begin{pmatrix} k_3\psi_{11} & k_3\psi_{01} & k_3\psi_{10} & k_3\psi_{00} \end{pmatrix}^{\mathrm{T}}. \tag{8}$$

Then, the rhs of (8) is duplicated via `EUDuplicationP` (as $s_6 = 2$) and

$$\bar{U}_6 = I_{6,\mathbb{V}} \times \bar{U}_6' \circ p_6 = \left(k_3\psi_{31j}p_{61ij} + k_3\psi_{30j}p_{60ij}\right)_{i,j=0,1}^{\mathrm{T}}, \tag{9}$$

where $\bar{U}_6'$ is equal to the duplicated version of the rhs of (8). The vector $\bar{U}_6$ has dimension four and its entries include two monomials of degree 3. Since the random vertex Y_5 is the unique parent of U_2 the `SymbolicExpectedUtility` function follows the same steps as before. `EUMultiSum` is called and

$$\bar{U}_5' \triangleq \bar{U}_6 +^{EU} \psi_2 = (h \cdot \bar{U}_6 + 1) \cdot k_2 \circ \begin{pmatrix} \psi_{21} & \psi_{20} & \psi_{21} & \psi_{20} \end{pmatrix}^{\mathrm{T}} + \bar{U}_6. \tag{10}$$

The polynomial $\bar{U}_5'$ is the sum of two monomials of degree 3 inherited from $\bar{U}_6$, of two monomials of degree 6 (from the first term on the rhs of (10)) and one monomial of degree 2 (from the last term on the rhs of (10)). Its dimension is equal to four since $c_6 = 4$ and $s_5 = 0$ (i.e. no `EUDuplicationPsi` is required). Thus, `EUMultiSum` manipulates the EU vector according to Proposition 2. The `EUMarginalization` function computes $\bar{U}_5 = I_{5,\mathbb{V}} \times ((\,\bar{U}_5'\ \bar{U}_5'\,)^{\mathrm{T}} \circ p_5)$. Each entry of $\bar{U}_5$ has twice the number of monomials of the entries of $\bar{U}_5'$ and each monomial of $\bar{U}_5$ has degree $d + 1$, where d is the degree of each monomial of $\bar{U}_5'$ (whose entries are homogeneous polynomials). These vectors also have the same dimension since $t_5 = 2$ and $r_5 = 2$. Thus, this `EUMarginalization` changes the EU vector according to Proposition 2. The entry $\bar{U}_5(y_3, y_4)$, with $y_3, y_4 = 0, 1$, of this EU can be shown to be equal to the sum of the terms in Table 2.

The algorithm then considers the controlled variable Y_4. Since $4 \not\in \mathbb{J}$, Y_4 is not the argument of a utility function with the highest index and therefore the algorithm calls the `EUMaximization` function. Suppose the DM decides to fix $Y_4 = 1$ when $Y_3 = 1$ and $Y_4 = 0$ when $Y_3 = 0$. Then `EUMaximization` returns $\bar{U}_4 = I_{4,\mathbb{D}} \times \bar{U}_5$, where $I_{4,\mathbb{D}}$ is a 2×4 matrix with ones in positions $(1, 1)$ and $(2, 4)$ and zeros otherwise. Proposition 2 is

Table 2 The utility funtion $\bar{U}_5$ is the sum of the three polynomials in this table

$k_2(\psi_{21}p_{51y_4y_3} + \psi_{20}p_{50y_4y_3})$
$k_3(\psi_{31y_4}p_{611y_4} + \psi_{30y_4}p_{601y_4})p_{51y_4y_3} + k_3(\psi_{31y_4}p_{610y_4} + \psi_{30y_4}p_{600y_4})p_{50y_4y_3}$
$hk_2k_3((\psi_{31y_4}p_{610y_4} + \psi_{30y_4}p_{600y_4})\psi_{20}p_{50y_4y_3} + (\psi_{31y_4}p_{611y_4} + \psi_{30y_4}p_{601y_4})\psi_{21}p_{51y_4y_3})$

respected since the entries of $\bar{U}_4$ have the same polynomial structure of those of $\bar{U}_5$ and $\bar{U}_4$ has dimension 2.

The `SymbolicExpectedUtility` function then applies in sequence the operations defined in Section 4.1. For the MID in Fig. 1 this sequentially computes the following quantities, assuming the DM fixed $Y_1 = 1$

$$\bar{U}_3' = h \cdot k_1 \cdot \bar{U}_4 \circ \psi_1 + \bar{U}_4 + k_1 \cdot \psi_1, \quad \bar{U}_3 = I_{3,\mathbb{V}} \times \left(\left(\bar{U}_3' \; \bar{U}_3' \; \bar{U}_3' \; \bar{U}_3' \right)^{\mathrm{T}} \circ p_3 \right),$$
$$\bar{U}_2 = I_{2,\mathbb{V}} \times \left(\bar{U}_3 \circ p_2 \right), \quad \bar{U}_1 = \left(1 \;\; 0 \right) \times \bar{U}_2.$$

The overall complexity of the algorithm can be formally deduced by counting the number of multiplications it involves. Given the number of such products for each of our EUoperations, the overall number of operations of Algorithm 4.2 is the sum of the multiplications of its operations and will depend on the topology of the ID network. Formally, letting

$$n^{\text{tot}} = \sum_{i \in [n]} \left(\mathbb{1}_{i \in \mathbb{V}} \; n_i^{\text{marg}} + \mathbb{1}_{i \in \mathbb{J}} \; n_i^{\text{sum}} \right),$$

where $\mathbb{1}$ denotes the indicator function, the overall complexity of our algorithm is $O(n^{\text{tot}})$.

4.4 Simulation study

To empirically investigate the complexity of the symbolic algorithm in Section 4.3, we perform a simulation study comprising of 5 IDs, whose features are summarized in Table 3 all with binary variables. We first produced a full symbolic definition of utilities and probabilities and then run our symbolic algorithm for both multiplicative and additive utility factorizations in MapleTH. This gives as output the EU vectors $\bar{U}_i$ associated to every random and decision nodes of the IDs. We also built the same networks using the GeNIe Modeler software of "BayesFusion, LLC" (freely available for academics at http://www.bayesfusion.com), which embeds numerical evaluation techniques for IDs. After building the networks in GeNIe, we specified numerical values for the probabilities and utilities and then ran the evaluation algorithm. It is important to highlight that GeNIe considers only additive factorizations between utility nodes.

The results of the study are summarized in Table 4. Whilst the memory allocated in MapleTH is almost identical for IDs with multiplicative and additive utility factorizations, the computation time as well as the number of monomials is much larger for MIDs. Comparing the computation times of GeNIe with those in MapleTH, we notice that whilst these are of the same magnitude for smaller IDs, for larger networks GeNIe appears to become significantly

Table 3 Summaries of the IDs considered in the simulation study: Net. - ID identifier; Free par. - number of free parameters; # $\mathbb{V}$ - number of random nodes; # $\mathbb{D}$ - number of decision nodes; m - number of utility nodes; # $E(G)^*$ - number of edges without those into decision nodes; Avg. indegree - average number of edges directed into vertices (without decision nodes)

Net.	Free par.	# $\mathbb{V}$	# $\mathbb{D}$	m	# $E(G)^*$	Avg. indegree
A	44	4	2	3	10	1.556
B	96	6	5	5	23	1.875
C	192	9	6	6	33	2.286
D	252	13	7	8	46	2.429
E	356	17	8	9	62	2.412

Table 4 Complexity summaries of the symbolic algorithms in MapleTH and of the numerical algorithms in GeNIe: Mem. All. - memory allocation; Time - computation time; # Mon. - number of monomials of the final EU vectors $\bar{U}_1$

	Multiplicative			Additive			GeNIe
Net.	Mem. All.	Time	# Mon.	Mem. All.	Time	# Mon.	Time
A	116KiB	3ms	84	114KiB	2ms	28	9.5ms
B	0.95MiB	36ms	1426	0.94MiB	29ms	138	13ms
C	1.05MiB	59ms	17810	1.04MiB	36ms	650	30ms
D	1.25MiB	1.45s	1590674	1.23MiB	82ms	17034	3.6s
E	1.41MiB	38.5s	$> 10^9$	1.38MiB	355ms	148106	81.2s

slower. However we underline that the two softwares produce different outputs: expected utility vectors with polynomial entries in MapleTH and numerical evaluation of the ID in GeNIe.

Although the efficiency of the symbolic algorithms highly depends on the size of the network, the simulation study in this section shows that even with the current capabilities of general-purpose computer algebra softwares symbolic techniques in decision making problems of medium/large scale are usefully applicable. In particular for IDs embedding additive factorizations, computation times increase at a slower pace than in the other cases (GeNIe and MIDs) and could thus be efficiently implemented in much larger domains than those presented here. However, it is uncommon to perform sensitivity studies over networks much larger than those investigated here. We refer a discussion of the handling of massive networks in our symbolic framework to Section 8.

5 Modifying the topology of the MID

Algorithm 4.2 works under the assumption that the MID is in extensive form whose importance was discussed in Section 3.2. It has been recognized that typically a DM will build an MID so that variables and decisions are ordered in the way they actually happen and this might not correspond to the order in which variables are observed. Thus, MIDs often are not in extensive form. But it is always possible to transform an MID into one in extensive form, although this might entail the loss of conditional independence structure. In Section 5.1 we consider two of the most common operations that can do this: edge reversal and barren node elimination.

In practice DMs often also include in the MID variables that subsequently turn out not to be strictly necessary for identifying an optimal policy. DMs are able to provide probabilistic judgements for conditional probability tables associated to an MID with variables describing the way they understand the unfolding of events. However their understanding usually includes variables that are redundant for the evaluation of the MID. In Section 5.2 we describe the polynomial interpretation of a criterion introduced in [42, 43] to identify a subgraph of the original MID whose associated optimal decision rule is the same as the one of the original MID.

5.1 Rules to transform an MID in extensive form

The two operations of **arc reversal** and **barren node removal** are often used in combination by first reversing the direction of some edges of the MID and then removing vertices that, consequently to the reversals, becomes barren, i.e. have no children [39].

Example 12 The MID on the left of Fig. 2 is a non-extensive variant of the MID in Fig. 1 not including the edge (Y_2, Y_4). The MID in the centre of Fig. 2 is obtained by the reversal of the edge (Y_2, Y_3) and the MID on the right of Fig. 2 is the network in extensive form obtained by deleting the barren node Y_2.

First we introduce a terminology to characterize a special pair of parent/child as in [11] for which edge reversals are simpler [36]. It is not the purpose of this paper to identify an optimal sequence of edge reversals, i.e. one yielding a simplified MID with the least number of vertices and edges. Instead we can use algorithms already devised for standard IDs to perform diagram transformations [40] by arc reversal and barren node removals. We say that Y_i is the father of Y_j and Y_j its son if the edge set of the MID includes (Y_i, Y_j) and there is no other path starting at Y_i and terminating at Y_j that connects them.

Example 13 For the MIDs in Fig. 2, both Y_4 and Y_5 are parents of Y_6, but only Y_5 is its father since there is the path (Y_4, Y_5, Y_6). Notice that a vertex can have only one father but more than one son.

Proposition 3 *The evaluation of an MID G provides the same optimal policy as the MID G' obtained by implementing any of the following manipulations:*

- ***Arc Reversal**: for $i, j \in \mathbb{V}$, if Y_i is the father of Y_j in G reverse the arc (Y_i, Y_j) into (Y_j, Y_i) and change the edge set as*

$$E(G') = E(G) \setminus \{(Y_i, Y_j)\} \cup \{(Y_k, Y_i) : k \in \{\Pi_j \cup j\} \setminus i\} \cup \{(Y_k, Y_j) : k \in \Pi_i\},$$

- ***Barren Node Removal**: for $i \in \mathbb{V}$, remove the vertex Y_i if this has no children and transform the diagram according to the following rules:*

$$V(G') = V(G) \setminus \{Y_i\}, \qquad E(G') = E(G) \setminus \big\{(Y_k, Y_i) : k \in \Pi_i\big\}.$$

Arc reversal and barren node removal change the symbolic parametrization of the MID according to Proposition 4. After an arc reversal, the diagram G' includes the edge (Y_j, Y_i) where $i < j$. Algorithm 4.2, and similarly the MapleTH function `SymbolicExpectedUtility`, works through a backward induction over the indices of the variables and, by construction, always either marginalize or maximize a vertex before its parents. It cannot therefore be applied straightforwardly to the diagram G'. We define here the adjusted Algorithm 4.2 which takes into account the reversal of an arc by, roughly speaking, switching the order in which the variables associated to the reversed edge are marginalized during the procedure. Specifically, in the adjusted Algorithm 4.2 a marginalization operation is performed over Y_i at the $n - j + 1$ backward inductive step, whilst for Y_j this happens at the $n - i + 1$ step. Therefore $\bar{U}'_j$ is the EU associated to G' after the

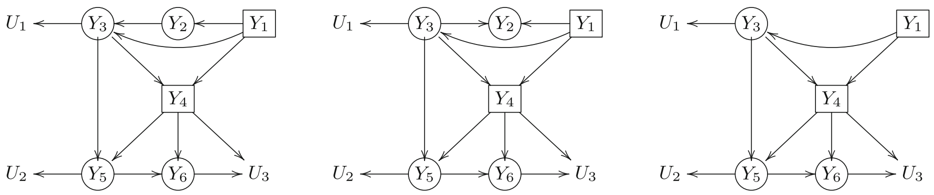


Fig. 2 Example of a sequence of manipulations of a non extensive form MID

marginalization of Y_i and $\bar{U}_i'$ is the EU after the marginalization of Y_j. Note that under this operation the sets $\mathbb{J}$ and B_i, $i \in [n]$, might change: we respectively call $\mathbb{J}'$ and B_i' the ones associated to G'.

Proposition 4 *Under the conditions of Proposition 3, let $p'_{iy\pi}$ and Π_i' be a parameter and a parent set associated to the diagram G' resulting from arc reversal and barren node removal:*

- *for $i, j \in \mathbb{V}$, if Π_i' and Π_j' are the parent sets of Y_i and Y_j after the reversal of the edge (Y_i, Y_j), then the parametrization associated to G' is*

$$p'_{iy_i\pi_i'} = \frac{p_{jy_j\pi_j} p_{iy_i\pi_i}}{\sum_{y_i \in \mathcal{Y}_i} p_{jy_j\pi_j} p_{iy_i\pi_i}}, \qquad p'_{jy_j\pi_j'} = \sum_{y_i \in \mathcal{Y}_i} p_{jy_j\pi_j} p_{iy_i\pi_i},$$

  for $\pi_i \in \mathcal{Y}_{\Pi_i}$, $\pi_j \in \mathcal{Y}_{\Pi_j}$, $\pi_i' \in \mathcal{Y}_{\Pi_i'}$, $\pi_j' \in \mathcal{Y}_{\Pi_j'}$, $y_i \in \mathcal{Y}_i$ and $y_j \in \mathcal{Y}_j$;
- *for $i, j \in \mathbb{V}$, assume that after the reversal of the edges (Y_i, Y_j), for every children Y_j of Y_i, Y_i is now a barren node and let $\Pi_{j\setminus i} = \Pi_j \setminus \{i\}$. Then*
  - *in the new parametrization p_i' is deleted;*
  - *in the old parametrization p_i is deleted and $p_{jy_j\pi_{j\setminus i}0} = \cdots = p_{jy_j\pi_{j\setminus i}r_i-1}$, for $y_j \in \mathcal{Y}_j$, $\pi_{j\setminus i} \in \mathcal{Y}_{\Pi_{j\setminus i}}$, where the fourth index of $p_{jy_j\pi_{j\setminus i}i}$, $i \in [r_i-1]$, refers to the instantiation of Y_i.*

The proof of this proposition is reported in Appendix A.3.

Example 14 Reversing the edge (Y_2, Y_3) in the MID on the left of Fig. 2, by Proposition 4 we obtain:

$$p'_{3y_3y_1} = p_{3y_31y_1}p_{21y_1} + p_{3y_30y_1}p_{20y_1}, \qquad p'_{2y_2y_3y_1} = \frac{p_{3y_3y_2y_1}p_{2y_2y_1}}{p_{3y_31y_1}p_{21y_1} + p_{3y_30y_1}p_{20y_1}}.$$

for $y_1, y_2, y_3 \in \{0, 1\}$. Proposition 4 also specifies that the deletion of the vertex Y_2 as on the right of Fig. 2 simply corresponds to cancelling the vectors p_2 and p_2' and setting $p_{3y_31y_1}$ equal to $p_{3y_30y_1}$ for any $y_1, y_3 \in \{0, 1\}$.

Note that arc reversals, just as posterior probabilities in symbolic inferences, transform EUs into rational functions of multilinear polynomials. However, Proposition 4 suggests a straightforward model's reparametrization, which maps EUs back to polynomial functions. In addition, Proposition 4 shows that manipulations of the diagram change the polynomial structure of the EUs under the new parametrization p' that we formally study in Lemmas 1 and 2 below. We assume here for simplicity that $i \notin P_j$, $j \in [m]$. There is no loss of generality in this assumptions since arguments of utility functions cannot be deleted from the diagram without changing the result of the evaluation.

Lemma 1 *Under the assumptions of Proposition 4 and in the notation of Theorem 1, suppose we reverse the arc (Y_i, Y_j) in an MID G. Let x be the smallest index in $\Pi_i \cup \Pi_j$. Evaluating G using the adjusted Algorithm 4.2 the following holds:*

1. *if $j \notin \mathbb{J}$, then*
   - *the entries of $\bar{U}'_j$ have $r_i r_{jba}/r_j \in \mathbb{Z}_{\geq 1}$*[7] *monomials of degree d_{jba}; for $i < k < j$, the entries of $\bar{U}'_k$ can have different polynomial structure from the ones of $\bar{U}_k$ according to Proposition 2;*
   - *the vectors $\bar{U}'_k$, $x < k \leq j$, have dimension $c_k = \prod_{s \in C_k \setminus \{k,\dots,n\}} r_s$ where $C_k = B_k \cup \{l : (Y_l, Y_i) \text{ or } (Y_l, Y_j) \in E(G')\}$;*
2. *if $j \in \mathbb{J} \cap \mathbb{J}'$, then*
   - *the entries of $\bar{U}'_j$ have $r_i r_{(j+1)ba}$ monomials of degree $d_{(j+1)ba} + 1$ and, for $i < k < j$, the entries of $\bar{U}'_k$ have a different polynomial structure from the ones of $\bar{U}_k$ according to Proposition 2;*
   - *for $x < k < j$, $\bar{U}'_k$ has dimension $a_k = \prod_{s \in \{A_k \setminus \{k,\dots,n\}\}} r_s$, with $A_k = C_k \cup P_{j_j}$;*
3. *if $j \notin \mathbb{J}'$, suppose $j \in P_t$ and s is the second highest index in P_t, then*
   - *for $s < k \leq j$, the entries of $\bar{U}'_k$ have the polynomial structure specified in point 2 and dimension a_k;*
   - *$i < k \leq s$, the entries of $\bar{U}'_k$ have the polynomial structure specified in point 1 and dimension c_k.*
   - *for $x < k \leq i$, $\bar{U}'_k$ has dimension c_k and the polynomial structure of its entries does not change;*

The proof of this lemma is provided in Appendix A.4.

We next consider how a barren node removal changes the EU vectors.

Lemma 2 *In the notation of Lemma 1, let Y_z be the child of Y_i in G with the highest index and remove the barren node Y_i in G'. Then*

- *for $i < k \leq z$, $\bar{U}'_k$ has c_k/r_i entries whose polynomial structure does not change;*
- *for $k \leq i$, $\bar{U}'_k$ has dimension c_k and its entries have r_{kba}/r_i monomials of degree $d_{kba} - 1$.*

The proof of this lemma is provided in Appendix A.4.

Example 15 After the reversal of the edge (Y_2, Y_3) from the network on the left of Fig. 2, the polynomial structure of the EUs associated to the original and manipulated diagrams is reported in Table 5 by $\bar{U}_i$ and $\bar{U}_i^r$ respectively. Since Y_3 is the only argument of U_1 we are in Item (2) of Lemma 1. The EU $\bar{U}_3^r$ is obtained running the adjusted Algorithm 4.2 over the graph in the centre of Fig. 2 after the marginalization of Y_2. This can be noted to change according to Lemma 1, by comparing its structure to the one of $\bar{U}_4$. Furthermore, $\bar{U}_2^r$ and $\bar{U}_1^r$ have the same polynomial structures as $\bar{U}_2$ and $\bar{U}_1$. The last 3 columns of the Table 5 show the polynomial structure of the EUs $\bar{U}_3^b$ associated to the MID on the right of Fig. 2 which does not include Y_2. According to Lemma 1, $\bar{U}_3^b$ has the same polynomial structure of $\bar{U}_3$ and for each row of the table, the number of monomials with degree d in $\bar{U}_1^b$ is half the number of monomials of $\bar{U}_1$ having degree $d + 1$.

[7]This is so since $r_{jba} = r' r_j$ for some $r' \in \mathbb{Z}_{\geq 1}$.

Table 5 Polynomial structure of the EUs for the original MID, $\bar{U}_j$, for the one after the reversal of the arc (Y_2, Y_3), $\bar{U}_j^r$ and for the one after the removal of the barren node Y_2, $\bar{U}_j^b$. The symbol # corresponds to the number of monomials, d. to the degree and s.f. to whether or not they are square free

$\bar{U}_2 \equiv \bar{U}_1$			$\bar{U}_3$			$\bar{U}_4$			$\bar{U}_3^r$			$\bar{U}_2^r \equiv \bar{U}_1^r$			$\bar{U}_3^b \equiv \bar{U}_1^b$		
#	d.	s.f.	#	d.	s.f.	#	d.	s.f.	#	d.	s.f.	#	d.	s.f.	#	d.	s.f.
4	4	✓	2	3	✓	2	3	✓	4	4	✓	4	4	✓	2	3	✓
8	5	✓	4	4	✓	4	4	✓	8	5	✓	8	5	✓	4	4	✓
16	6	✓	8	5	✓	4	7	✓	8	8	✓	16	6	✓	8	5	✓
8	8	✓	4	7	✓							8	8	✓	4	7	✓
32	9	✓	16	8	✓							32	9	✓	16	8	✓
16	12	✗	8	11	✗							16	12	✗	8	11	✗

5.2 The sufficiency principle

After an MID has been transformed in extensive form according to the rules in Section 5.1, further manipulations can be applied to simplify its evaluation, such as the sufficiency principle, which mirrors the concept of sufficiency in statistics and is based on the concept of d-separation for DAGs [37] formally defined below.

We first introduce a few concepts from graph theory. The moralized graph G^M of the MID G is a graph with the same vertex set of G. Its directed edges include the directed edges of G and an undirected edge between any two vertices which are not joined by an edge in G but which are parents of the same child in Y_i, $i \in \mathbb{V}$. The skeleton of G^M, $\mathcal{S}(G^M)$, is a graph with the same vertex set of G^M and an undirected edge between any two vertices $(Y_i, Y_j) \in V(G^M)$ if and only if there is a directed or undirected edge between Y_i and Y_j in G^M.

Definition 7 For any three disjoint subvectors $Y_A, Y_B, Y_C \in V(G^M)$, Y_A is *d-separated* from Y_C by Y_B in the moralized graph G^M of an MID G if and only if any path from any vertex $Y_a \in Y_A$ to any vertex $Y_c \in Y_C$ passes through a vertex $Y_b \in Y_B$ in its skeleton $\mathcal{S}(G^M)$.

Proposition 5 *Let $j \in \mathbb{D}$, $i \in \mathbb{V} \cap \Pi_j$ and Ch_i be the index set of the children of Y_i. Then if Y_i is d-separated from $\{U_k$, for k s.t. $i \leq j_k\}$ by $\{Y_k : k \in \{\Pi_j \setminus i\}\} \cup \{Y_k : k \in \mathbb{D}\}$ in the MID G, the sufficiency principle guarantees that the evaluation of the graph G' provides the same optimal policy as G, where G' is such that $V(G') = V(G) \setminus \{Y_i\}$ and $E(G')$ is equal to*

$$E(G) \setminus \{(Y_i, Y_j) : j \in Ch_i\} \setminus \{(Y_k, Y_i) : k \in \Pi_i\} \cup \{(Y_k, Y_j) : j \in Ch_i, k \in \Pi_i\}.$$

The sufficiency principle can be equally stated for a vector of variables [42, 43]. However, we can simply apply the criterion in Proposition 5 for each variable of the vector and obtain the same result.

Example 16 The MID in Fig. 1 is already moralized. Any path from Y_2 into U_i, $i \in [3]$, goes through both Y_3 and Y_4. By Proposition 5, we can delete Y_2 and the modified diagram is given on the right of Fig. 2. This happens to be equal to the diagram resulting from the reversal of the arc (Y_2, Y_3) and the deletion of Y_2.

We now formalize how this principle changes our parametrization.

Proposition 6 *Let $i, j, k \in \mathbb{V}$ and G be an MID. Let Y_i be a vertex removed after the application of the sufficiency principle to G and G' the obtained MID. Assume Y_i to be the father of Y_k and a parent (not the father) of Y_j in G and let Π_k' be the parent set of a vertex Y_k in G'. Then the reparametrization of the MID with graph G' is*

$$p'_{ky_k\pi_k'} = \sum_{y_i \in \mathcal{Y}_i} p_{ky_k\pi_k} p_{iy_i\pi_i},$$

$$p'_{jy_j\pi_j'} = \sum_{y_j \in \mathcal{Y}_j} p_{jy_j\pi_j} \frac{\prod_{l \in \Pi_j \setminus [i-1]} \sum_{\mathcal{Y}_{\Pi_l \cap \Pi_k \cap \Pi_i}} p_{ly_l\pi_l} p_{iy_i\pi_i}}{\sum_{y_i \in \mathcal{Y}_i} \prod_{l \in \Pi_j \setminus [i-1]} \sum_{\mathcal{Y}_{\Pi_l \cap \Pi_k \cap \Pi_i}} p_{ly_l\pi_l} p_{iy_i\pi_i}}$$

The proof of this proposition is provided in Appendix A.5. Again, this new parametrization p' implies a change in the EU vectors.

Lemma 3 *Assume the vertex Y_i is removed using the sufficiency principle and that Y_j is the child of Y_i with the highest index. Under the notation of Theorem 1 the EU vectors in G' are such that*

1. *for $k < i$, the entries of $\bar{U}_k'$ have r_{kba}/r_i monomials of degree $d_{kba} - 1$, whilst for $k > i$ their structure does not change.*
2. *for $k \leq j$, $\bar{U}_k$ has now dimension $\prod_{s \in C_k} r_s$, where $C_k = B_k \cup \Pi_i \setminus \{k, \dots, n\}$, whilst for $k > j$ its dimension does not change.*

Proof Item 1 of Lemma 3 is a straightforward consequence of Proposition 2, since the deletion of the vertex Y_i entails one less EUMarginalization during Algorithm 4.2. Item 2 of Lemma 3 follows from the fact that the sets B_k and C_k only affect the dimension of the EU vectors. □

Since the application of the sufficiency principle to the diagram of Fig. 1 provides the same output network as the one obtained from the reversal of the edge (Y_2, Y_3) and the removal of Y_2, an illustration of these results can be found in Table 5.

6 Asymmetric decision problems

The new symbolic representation of decision problems we introduce here enables us to concisely express a large amount of information that might not be apparent from an ID. Different types of extra information, often consisting of asymmetries of various kinds, have been explicitly modelled in graphical extensions of the ID model [3, 4, 20, 28, 41] and are found in the descriptions of many applied decision problems. Although providing a framework for the evaluation of more general decision problems, many of these extensions lose the intuitiveness and the simplicity associated with IDs. Within our symbolic approach we are able to elegantly and concisely characterize asymmetric decision problems through manipulations of the polynomials representing the ID's EU as we show next.

Asymmetries can be categorized in three classes. If the possible outcomes or decision options of a variable vary depending on the past, the asymmetry is called *functional*. If the very occurrence of a variable depends on the past, the asymmetry is said to be *structural*. *Order* asymmetries are present if $\{Y_i : i \in \mathbb{D}\}$ is not totally ordered. In this section we only deal with functional asymmetries. Heuristically, for a functional asymmetry the observation of y_A, $A \subset [n]$, restricts the space $\mathcal{Y}_B$ associated to a vector Y_B, such that $A \cap B = \emptyset$. This new space, $\mathcal{Y}'_B$ say, is a subspace of $\mathcal{Y}_B$.

In Theorem 2 we characterize an asymmetry between two chance nodes and, depending on the stage of the evaluation, this may entail setting equal to zero monomials in either some or all the rows of the EU vector. We present the result for elementary asymmetries of the following form: if $Y_i = y_i$ then $Y_j \neq y_j$. Composite asymmetries are unions of simple asymmetries and the features of the EU vectors in more general cases can be deduced through a sequential application of Theorem 2.

Theorem 2 *Let G be an MID, Y_i and Y_j be two random variables with $j > i$, U_x be the utility node following Y_j in the DS. Assume the asymmetry $Y_i = y_i \Rightarrow Y_j \neq y_j$ holds and that k and z are the highest indices such that $j \in B_k$ and $i \in B_z$ and assume $k > j$. Then*

- *for $j < t \leq z$, $\bar{U}_t$ has $\prod_{s\in B_t\setminus\{i\cup j\}} r_s$ rows with no monomials;*
- *for $i < t \leq j$, $\bar{U}_t$ has $\prod_{s\in B_t\setminus\{i\}} r_s$ rows with polynomials all with a different structure. Specifically, these consists, in the notation of Theorem 1, of s_{tba} monomials of degree d_{tba}, where, for $a = x, \dots, m$ and $b = l, \dots, a$,*

$$s_{tba} = \left(\binom{a-x}{b-l} - 1\right)\prod_{s=t}^{j_a} r_s/r_j;$$

- *for $t \leq i$, each row of $\bar{U}_t$ has in the notation of Theorem 1, f_{tba} monomials of degree d_{tba}, where for $a = x, \dots, m$ and $b = l, \dots, a$*

$$f_{tba} = \left(\binom{a-x}{b-l} - 1\right)\prod_{s=t}^{j_a} r_s/(r_j \cdot r_i).$$

The proof of this theorem is provided in Appendix A.6. Corollary 2 gives a characterization of simple asymmetries between any two variables, whether they are controlled or non-controlled. This follows from Theorem 2 since controlled variables can be thought of as a special case of random ones.

Corollary 2 *In the notation of Theorem 1 and under the assumptions of Theorem 2, with the difference that Y_i and Y_j are two variables, controlled or non-controlled, we have that*

- *for $j < t \leq z$, each row of $\bar{U}_t$ has $\prod_{s\in B_t\setminus\{i\cup j\}} r_s$ rows with no monomials;*
- *for $i < t \leq j$, $\bar{U}_t$ has at most $\prod_{s\in B_t\setminus\{i\}} r_s$ rows with polynomials all with a different structure. Specifically, these consists of between s_{tba} and r_{tba} monomials of degree d_{tba}, for $a = x, \dots, m$ and $b = l, \dots, a$;*
- *for $t \leq i$, some rows of $\bar{U}_t$ have a number of monomials of degree d_{tba} between f_{tba} and r_{tba}, for $a = x, \dots, m$ and $b = l, \dots, a$.*

Example 17 (Example 4 continued) Assume that the DM believes the decision problem is characterized by three composite asymmetries:

- if Y_1 was fixed to 1, then $Y_4 = 1$ cannot be chosen;
- if either Y_2 or Y_3 were observed to be equal to 1 then $Y_5 = 1$;
- if $Y_4 = 1$ then both Y_5 and Y_6 are equal to 1.

A graphical representation of these asymmetries is given in Fig. 3, in the form of a sequential influence diagram [28]. Asymmetries are represented as labels on new dashed arcs. If the asymmetry is composite, then vertices can be grouped through a dashed ellipse and dashed arcs can either start or finish by the side of these ellipses. Although this generalization of the MID in Fig. 1 graphically captures the asymmetries, most of its transparency is now lost. Instead asymmetries have the opposite effect on our polynomial representation of MIDs by greatly simplifying the structure of the EUs.

In this asymmetric framework the first row of $\bar{U}_6$ corresponds to $k_3\psi_{311}p_{6111}$, whilst its second row is empty. This is because according to Theorem 2 the monomial $k_3\psi_{301}p_{6011}$ in (9) is cancelled by the asymmetry $Y_4 = 1 \Rightarrow Y_6 = 1$, $k_3\psi_{311}p_{6101}$ by $Y_4 = 1 \Rightarrow Y_5 = 1$ and $k_3\psi_{301}p_{6001}$ by both asymmetries. The imposition of asymmetries further reduces from ten to three the number of monomials in $\bar{U}_5$ which becomes

$$k_3\psi_{311}p_{6111}p_{511i} + k_2\psi_{21}p_{511i} + hk_2k_3\psi_{311}\psi_{21}p_{6111}p_{511i}, \qquad i = 0, 1.$$

Suppose the DM decided to fix $Y_4 = 0$ if $Y_3 = 1$ and $Y_4 = 1$ if $Y_3 = 0$. The entry of $\bar{U}_3$ for which $Y_2 = 1$ and $Y_1 = 1$ can be written as the sum of the terms

$$(k_2\psi_{21} + k_3\psi_{311}p_{6111}(1 + kk_2\psi_{21}))p_{5110}p_{3011} + k_1(\psi_{10}p_{3011} + \psi_{11}p_{3111})$$
$$kk_1k_3\psi_{11}p_{5101}p_{3111}((1 + k_2\psi_{21})(\psi_{300}p_{6010} + \psi_{310}p_{6110})).$$

This polynomial consists of only nine monomials. This compared with the number of monomials in the symmetric case, 42 (see Table 5), means that even in this small problem the number of monomials is decreased by over three quarters.

So the example above illustrates that under asymmetries the polynomial representation is simpler than standard methods but still able to inform decision centres about the necessary parameters to elicit. A more extensive discussion of the advantages of symbolic approaches in asymmetric contexts, although fully inferential ones, can be found in [24]. Finally it is

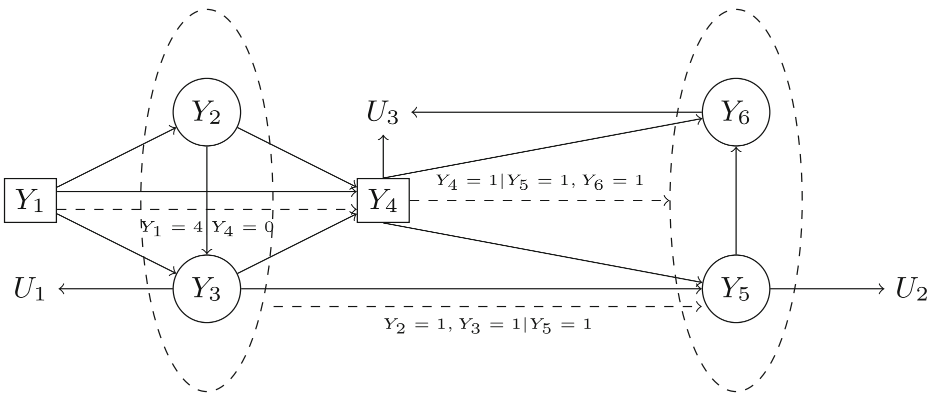


Fig. 3 Representation of the asymmetric version of the MID of Fig. 1 through a sequential influence diagram

possible to develop a variant of Algorithm 4.2 which explicitly takes into account the asymmetries of the problem *during* the computation of the EU vectors. Note that this approach would be computationally more efficient, since this would require the computation of a smaller number of monomials/polynomials.

7 An example

In this section we study the polynomial features of the EUs associated to the MID in Fig 1. We focus on the selection of the decision variable Y_4 and consider two different scenarios in which the DM provides two different sets of information of the relevant parameters. In the first scenario the elicitation is complete, i.e. for each parameter the DM delivers the unique numerical value specified in the left hand side of Table 6. The second scenario combines unique probability specifications, symbolic parameters and qualitative information. Specifically the DM does not elicit p_{5111}, p_{6001}, p_{6010}, p_{6011} and ψ_{301}, because e.g. there is strong uncertainty related to their values, specifies the two relationships $p_{5111} = p_{6011}$ and $p_{6001} = p_{6010}$ and assigns specific values to the remaining parameters as indicated in Table 6.

In the first scenario, using any standard propagation algorithm or by simply substituting the appropriate numerical values from Table 6 into the EU polynomial $\bar{U}_5(y_4, y_3)$ from Table 2, the DM would be suggested to choose $Y_4 = 1$ if $Y_3 = 0$ and $Y_4 = 0$ if $Y_3 = 1$, since

$$\bar{U}_5(1,0) = 0.4465,\ \bar{U}_5(0,0) = 0.4460 \text{ and } \bar{U}_5(1,1) = 0.3074,\ \bar{U}_5(0,1) = 0.3755.$$

In an automated decision making process the DM might overlook the small difference in EU values when $Y_3 = 0$, which already suggests that small changes in parameters' values may lead to different preferred policies.

A symbolic study of EUs in the partial elicitation case of the second scenario can provide insights on why the DM's decision making may not be robust. Substituting the partial numeric specification in Table 6 into the polynomial from Table 2 yields

$$\begin{aligned}\bar{U}_5(y_4, y_3) \;&=\; 0.2p_{50y_4y_3} + 0.4\left(\psi_{31y_4}p_{611y_4} + \psi_{30y_4}p_{601y_4}\right)p_{51y_4y_3}\\ &\quad+\; 0.472\left(\psi_{31y_4}p_{610y_4} + \psi_{30y_4}p_{600y_4}\right)p_{50y_4y_3}\end{aligned}$$

Table 6 Complete and partial specification of the parameters associated to MID in Fig. 1 for the optimization step over Y_4. By the sum-to-one condition $p_{6001} = p_{6010}$ is equivalent to $p_{6101} = p_{6110}$

Parameters' specifications

Complete				*Partial*			
$p_{6111} = 0.3$,	$p_{5110} = 0.2$,	$\psi_{311} = 0$,	$k_3 = 0.4$	$p_{6100} = 0.3$	$\psi_{311} = 0$,	$\psi_{21} = 0$,	$k_3 = 0.4$,
$p_{6110} = 0.2$,	$p_{5101} = 0.9$,	$\psi_{310} = 0.4$,	$k_2 = 0.2$	$p_{5110} = 0.2$,	$\psi_{310} = 0.4$	$\psi_{20} = 1$,	$k_2 = 0.2$,
$p_{6101} = 0.2$,	$p_{5100} = 0.6$,	$\psi_{301} = 0.8$,	$k_1 = 0.2$	$p_{5101} = 0.9$,	$\psi_{300} = 1$,		$k_1 = 0.2$,
$p_{6100} = 0.3$,	$p_{5111} = 0.7$,	$\psi_{300} = 1$,	$h = 0.9$	$p_{5100} = 0.6$,			$h = 0.9$
$\psi_{21} = 0$,		$\psi_{20} = 1$,		$p_{5111} = 1 - p_{6111}$,		$p_{6001} = p_{6010}$,	

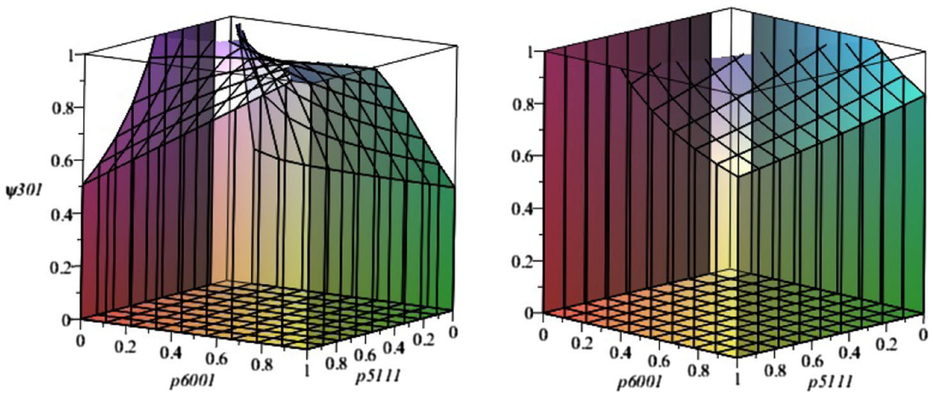


Fig. 4 Admissible domains, expressed in the unknowns ψ_{301}, p_{6001} and p_{5111}, for the combinations of parameters leading to a preferred decision $Y_4 = 0$ (coloured regions) and $Y_4 = 1$ (white regions) for the MID in Fig. 1 given the partial numeric specification in Table 6

which is further specialised in

$$\begin{aligned}
\bar{U}_5(1,1) &= 0.2(1-p_{5111}) + 0.4\psi_{301}p_{5111}^2 + 0.472\psi_{301}p_{6001}(1-p_{5111})\\
\bar{U}_5(0,1) &= 0.100976 + 0.6192p_{6001}\\
\bar{U}_5(1,0) &= 0.16 + 0.08\psi_{301}p_{5111} + 0.3776p_{6001}\\
\bar{U}_5(0,0) &= 0.233984 + 0.6192p_{6001}
\end{aligned}$$

Under the partial specification scenario, the admissible domains when $Y_3 = 1$, namely $\arg\max\{\bar{U}_5(1,1), \bar{U}_5(0,1)\}$, are reported on the left hand side of Fig. 4 and the associated *indifference surface* is defined by the equation $\bar{U}_5(1,1) = \bar{U}_5(0,1)$. The right hand side of Fig. 4 shows the admissible domains when $Y_3 = 0$. For $Y_3 = 1$, the combination of values elicited by the DM is well inside the colored region, whilst for $Y_3 = 0$ it is very closed to the indifference surface defined by the points where the DM is indifferent between the two policies. The indifference surface for $Y_3 = 0$ is very smooth and regular since the associated variety is defined by a simple multilinear polynomial. Conversely, the surface for $Y_3 = 1$ exhibits more interesting features since the associated variety is defined by a quadratic function.

Additional information about the DM's decision problem can be gained by investigating the admissible domains defined by two parameters only, when the third one is fixed to the value chosen in the complete elicitation scenario. In Fig. 5 we report the regions for $Y_3 = 0$. The admissible domains are very "smooth" and the indifference surfaces are all monotonic functions. In all the plots the complete elicitation point is very close to the indifference curve and thus small perturbations of the parameters can lead to a different preferred policy. Much more robust is the DM's potential decision in the case $Y_3 = 1$, since all the complete elicitation points are well inside the admissible domains (reported in Fig. 6). Note how in this

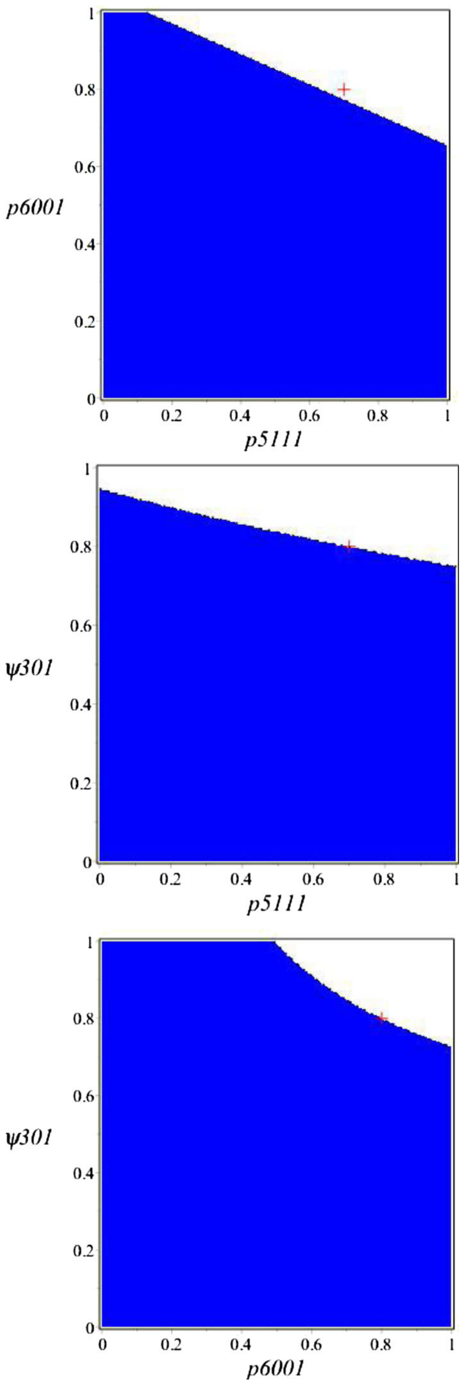
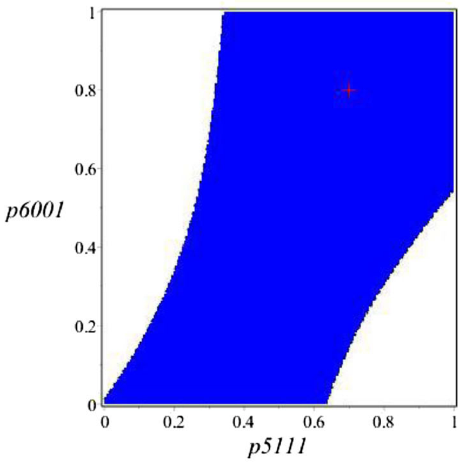
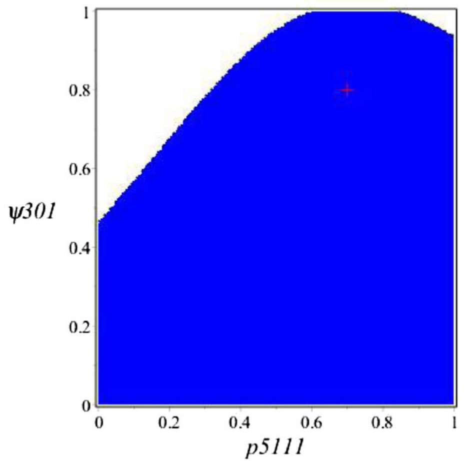


Fig. 5 Admissible domains for subsets of 2 elements of the parameter space for $Y_3 = 0$, fixing the third to the value of the complete elicitation (colored for $Y_4 = 0$ and white for $Y_4 = 1$)





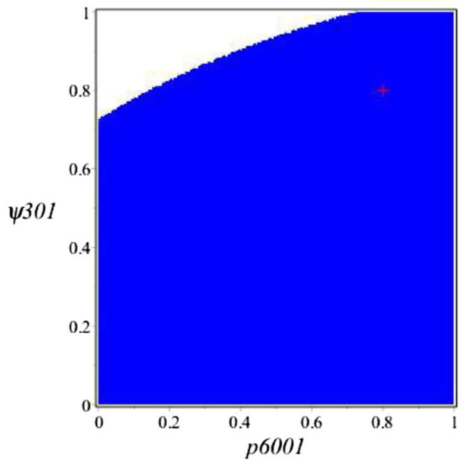


Fig. 6 Admissible domains for subsets of 2 elements of the parameter space for $Y_3 = 1$, fixing the third to the value of the complete elicitation (colored for $Y_4 = 0$ and white for $Y_4 = 1$)

case the admissible domains have a much more complex geometry, due to the polynomial structure of the indifference surface variety. We highlight a few points from this example:

- the geometry of the admissible regions has provided insights on the decision making process. In much more complex problems, tools of algebraic geometry [16] can still be used to guide DMs and to uncover even more surprising features;
- although other approaches allow for non exact probability specifications, our symbolic characterization provides a straightforward platform to input qualitative information, e.g. equality of two parameters, which entails a simple reparametrization of the problem;
- the symbolic approach is particularly efficient for this type of sensitivity studies since a DM can simply plug-in different combinations of values for the unknowns and instantaneously observe the results. In full numerical domains, the propagation of EUs would need to performed for each combination of values and this can become computationally very expensive;
- if, after robustness studies as the one in this example, the DM is still not convinced about a preferred course of action, our algorithm can be adapted to run separately for each admissible domain back to the root of the MID. In this way it would then output the admissible regions for each multivariate available policy together with its defining polynomial;
- the identification of the admissible domains consists of the solution of a system of polynomial inequalities. We are currently investigating these domains using semi-algebraic methods [5];
- all these methods are especially informative in asymmetric domains, since different policies can be associated to polynomial having very different properties. This is because, as shown in Theorem 2, in contast to standard MIDs, different policies can be associated with very differently structured polynomials.

8 Discussion

With this work we have developed symbolic methods - currently being successfully applied to the analysis of probabilistic graphical models - to study MIDs. We have defined a complete toolkit to deal with standard operations for MIDs from a symbolic point of view, such as the computations of EUs, possible manipulations of the diagram and asymmetries. Whilst in open-loop analyses our symbolic definition finds its natural application, in closed-loop analyses the EU-Maximization operation becomes critical. In some specific cases, as illustrated through the example in Section 7, partial parameters' elicitations will allow the DM to perform such step. In more general cases we still need to formalize such maximization techniques for example by adopting semi-algebraic methods which have already proved successful in other applications [5]. We expect these to be particularly useful in asymmetric domains, since different polynomial structures can inform even more deeply the DM about the structure of the decision space.

We here provide a full report of an implementation of our methodology within an accessible computer algebra system. Of course when addressing very large problems generic tools can have difficulties handling the number of unknown variables that need to be stored in the computer memory and computations may become infeasible. However, there are ways around this memory problem. For example by imposing certain conditions on the model - formally discussed in [34] - computations can then be distributed. This can dramatically reduce complexity and make calculations again feasible albeit with the necessary addition of further software - designed for the particular application - which intelligently merges together the outputs of the different contributing distributed components of the system: see also [45]. The simulations carried out in Section 4.4 and the example in Section 7 showed that the methodology and algorithm presented in this paper are competitive and allow for the analysis of more general classes of models than traditional methods. Implementations based on specialised programs rather than on a general purpose software like MapleTH will enable the analysis of more complex MIDs.

Also in the case the variables take values in continuous spaces, EU exhibits a similar polynomial representation to the one discussed in this paper for discrete variables. In the continuous case the unknown quantities of the polynomials are low order moments. Examples of these polynomials are presented in [34]. Just as in the discrete case, the manipulations of the diagrams for policies with continuous variables and their associated asymmetries can be described as operations over the polynomials. A full study of the symbolic representation of EUs in a continuous domain will be reported in future work.

Appendix A Proofs

A.1 Proof of proposition 1

We develop the proof via backward induction over the random and decision vertices of the MID, starting from Y_n. Define, for $i \in [n]$,

$$\hat{U}_i = \int_{\mathcal{Y}_{[n]^{\mathbb{V}}_{i-1}}} \max_{\mathcal{Y}_{[n]^{\mathbb{D}}_{i-1}}} \sum_{I \in \mathcal{P}_0([m])} h^{n_I - 1} \prod_{i \in I} k_i U_i(y_{P_i}) f\left(y_{[n]^{\mathbb{V}}_{i-1}} \mid y_{[i-1]}\right) \mathrm{d}y_{[n]^{\mathbb{V}}_{i-1}},$$

where $[n]^{\mathbb{V}}_{i-1} = [n] \setminus [i-1] \cap \mathbb{V}$, $[n]^{\mathbb{D}}_{i-1} = [n] \setminus [i-1] \cap \mathbb{D}$ and $\Pi_{[n]^{\mathbb{V}}_{i-1}} = \cup_{j \in [n]^{\mathbb{V}}_{i-1}} \Pi_j$. The quantity $\hat{U}_i$ corresponds to an overall EU score after having marginalized/maximized $Y_i, \dots, Y_n$.

The DM's preferences are a function of Y_n only through $k_m U_m(y_{P_m})$, since by construction $n = j_m \in \mathbb{J}$. Therefore this quantity can be either maximized or marginalized as in (3) to compute $\bar{U}_n(y_{B_n})$. Note that B_n includes only the indices of the variables $\bar{U}_n$ formally depends on, since $B_n = P_m \setminus \{n\}$, if $n \in \mathbb{D}$, whilst $B_n = P_m \cup \Pi_n \setminus \{n\}$, if $n \in \mathbb{V}$. Then

$$\hat{U}_n = \sum_{I \in \mathcal{P}_0([m])} h^{n_I - 1} \prod_{i \in I} \left(\mathbb{1}_{\{i \neq n\}} \left[k_i U_i(y_{P_i})\right] + \mathbb{1}_{\{i = n\}} \left[\bar{U}_i(y_{B_i})\right]\right).$$

Now consider Y_{n-1}. If $n-1 \notin \mathbb{J}$, then $\hat{U}_n$ is a function of Y_{n-1} only through $\bar{U}_n$. Therefore maximization and marginalization steps can be computed as in (5) to compute $\bar{U}_{n-1}(y_{B_{n-1}})$. Again B_{n-1} includes the indices of the variables $\bar{U}_{n-1}$ formally depends on, since $B_{n-1} = P_m \setminus \{n, n-1\}$, if $n, n-1 \in \mathbb{D}$, $B_{n-1} = P_m \cup \Pi_n \cup \Pi_{n-1} \setminus \{n, n-1\}$, if $n, n-1 \in \mathbb{V}$, $B_{n-1} = P_m \cup \Pi_{n-1} \setminus \{n, n-1\}$, if $n \in \mathbb{D}$ and $n-1 \in \mathbb{V}$, $B_{n-1} = P_m \cup \Pi_n \setminus \{n, n-1\}$, if $n \in \mathbb{V}$ and $n-1 \in \mathbb{D}$. Then

$$\hat{U}_{n-1} = \sum_{I \in \mathcal{P}_0([m])} h^{n_I - 1} \prod_{i \in I} (\mathbb{1}_{\{i \neq n\}} k_i U_i(y_{P_i}) + \mathbb{1}_{\{i = n\}} \bar{U}_{i-1}(y_{B_{i-1}})).$$

Conversely, if $n-1 \in \mathbb{J}$, $\hat{U}_n$ is potentially a function of Y_{n-1} through both $U_{m-1}(y_{P_{m-1}})$ and $\bar{U}_n(y_{B_n})$ and note that $\hat{U}_n$ can be written in this case as

$$\hat{U}_n = \sum_{I \in \mathcal{P}_0([m-2])} h^{n_I - 1} \prod_{i \in I} k_i U_i(y_{P_i}) + U'_{m-1} + \left(\sum_{i \in \mathcal{P}_0([m-2])} h^{n_i - 1} \prod_{i \in I} k_i U_i(y_{P_i}) \right) U'_{m-1},$$

where

$$U'_{m-1} = hk_{m-1} U_{m-1}(y_{P_{m-1}}) \bar{U}_n(y_{B_n}) + k_{m-1} U_{m-1}(y_{P_{m-1}}) + \bar{U}_n(y_{B_n}).$$

Therefore optimization and marginalization steps can be performed over U'_{m-1} as specified in the two (4) respectively. Then note that $\hat{U}_{n-1}$ can be written as

$$\begin{aligned}\hat{U}_{n-1} &= \sum_{I \in \mathcal{P}_0([m-2])} h^{n_I - 1} \prod_{i \in I} k_i U_i(y_{P_i}) + \bar{U}_{n-1}(\cdot) + \left(\sum_{i \in \mathcal{P}_0([m-2])} h^{n_i - 1} \prod_{i \in I} k_i U_i(y_{P_i}) \right) \bar{U}_{n-1}(\cdot) \\ &= \sum_{I \in \mathcal{P}_0([m-1])} h^{n_I - 1} \prod_{i \in I} (\mathbb{1}_{\{i \neq n-1\}} k_i U_i(y_{P_i}) + \mathbb{1}_{\{i = n-1\}} \bar{U}_i(y_{B_i})).\end{aligned}$$

Now for a $j \in [n-2]$ and assuming with no loss of generality that k is the index of a utility vertex such that $j_{k-1} < j \leq j_k$, we have that

$$\hat{U}_j = \sum_{I \in \mathcal{P}_0([k])} h^{n_I - 1} \prod_{i \in I} (\mathbb{1}_{\{i \neq j\}} k_i U_i(y_{P_i}) + \mathbb{1}_{\{i = j\}} \bar{U}_i(y_{B_i})).$$

Therefore at the following step, when considering Y_{j-1}, we can proceed as done with Y_{n-1} by maximization and marginalization in (4)–(5) to compute $\hat{U}_{j-1}$. Thus at the conclusion of the procedure, $\bar{U}_1$ yields the EU of the optimal decision.

A.2 Proof of theorem 1

For a subset $I \in \mathcal{P}_0([m])$, let j_I be the index of the variable appearing before the utility vertex with index $U_{\max_I}$ in the decision sequence. Let $C_{i,I} = \{z \in \mathbb{V} : i \leq z \leq j_I\}$ and recall that l is the index of the first utility node following Y_i in the DS. The EU function of (3)–(5) can be (less intuitively) written as $\bar{U}_i(y_{B_i}) = \sum_{I\in\mathcal{P}_0(\{l,\dots,m\})} \bar{U}_{i,I}(y_{B_i})$, where $\bar{U}_{i,I}(y_{B_i})$ is defined as

$$\bar{U}_{i,I}(y_{B_i}) = \sum_{I\in\mathcal{P}_0(\{l,\dots,m\})} h^{n_I-1}\prod_{s\in I} k_s U_s(y_{P_s}) \sum_{y_{C_{i,I}}\in\mathcal{Y}_{C_{i,I}}} \prod_{j\in C_{i,I}} P(y_j|y_{\Pi_j}). \tag{11}$$

The EU therefore depends on the power set of the indices of the utility vertices subsequent to Y_i in the decision sequence. We can note that for any $I, J \in \mathcal{P}(\{l,\dots,m\})$ such that $\#I = \#J$ and $U_{\max_I} = U_{\max_J}$, $\bar{U}_{i,I}(y_{B_i})$ and $\bar{U}_{i,J}(y_{B_i})$ have the same polynomial structure since $C_{i,I} = C_{i,J}$. Now for $a = l,\dots,m$ and $b = l,\dots,a$, by the properties of binomial coefficients, $\binom{a-l}{b-l}$ counts the number of elements $I \in \mathcal{P}_0(\{l,\dots,m\})$ having $\#I = b-l+1$ and including a. Thus r_{iba} in (7) counts the correct number of monomials having a certain degree since $\mathcal{Y}_{C_{i,I}} = \times_{t\in C_{i,I}}\mathcal{Y}_t$. Further note that considering each combination of b and a in the ranges specified above, we count each element of $\mathcal{P}_0(\{l,\dots,m\})$.

By having a closer look at d_{iba} in (7) it is easy to deduce the corresponding degree of these monomials. The first term of d_{iba}, $(b-l)$, computes the degree associated to the criterion weight h, since $b-l = n_I - 1$ and the second term, $2(b-l+1)$, computes the degree associated to the product between the criterion weights k_s and the utilities $U_s(y_{P_s})$ for $s \in C_{i,I}$. The last term w_{ia} corresponds to the degree deriving from the probabilistic part of (11), which is equal to the number of non-controlled vertices between Y_i and $Y_{j_{\max_I}}$ (both included).

Since the set B_i includes the arguments of $\bar{U}_i(y_{B_i})$ and $\mathcal{Y} = \times_{i\in[n]}\mathcal{Y}_i$, (6) guarantees that the dimension of the EU vector is $\prod_{t\in B_i} r_t$.

A.3 Proof of proposition 4

After the reversal of the arc (Y_i, Y_j) into (Y_j, Y_i), the new parent sets of these two variables are $\Pi'_j = \{\Pi_j \cup \Pi_i \setminus i\}$ and $\Pi'_i = \{j \cup \Pi_i \cup \Pi_j \setminus i\}$. Call $\Pi_{j\setminus i} = \{\Pi_j \setminus i\}$. It then follows that

$$\begin{aligned}
p_{iy_i\pi'_i} &= P(y_i \mid y_{\Pi'_i}) = P(y_i \mid y_{\Pi_{j\setminus i}}, y_{\Pi_i}, y_j) = \frac{P(y_j \mid y_{\Pi_{j\setminus i}}, y_{\Pi_i}, y_i)P(y_i \mid y_{\Pi_{j\setminus i}}, y_{\Pi_i})}{P(y_j \mid y_{\Pi_{j\setminus i}}, y_{\Pi_i})}\\
&= \frac{P(y_j \mid y_{\Pi_j})P(y_i \mid y_{\Pi_i})}{P(y_j \mid y_{\Pi_{j\setminus i}}, y_{\Pi_i})} = \frac{P(y_j \mid y_{\Pi_j})P(y_i \mid y_{\Pi_i})}{\sum_{y_i\in\mathcal{Y}_i} P(y_j \mid y_i, y_{\Pi_{j\setminus i}})P(y_i \mid y_{\Pi_i})}\\
&= \frac{p_{jy_j\pi_j}\,p_{iy_i\pi_i}}{\sum_{y_i\in\mathcal{Y}_i} p_{jy_j\pi_j}\,p_{iy_i\pi_i}},
\end{aligned}$$

and

$$\begin{aligned}
p'_{jy_j\pi'_j} &= P(y_j \mid y_{\Pi'_j}) = P(y_j \mid y_{\Pi_{j\setminus i}}, y_{\Pi_i}) = \sum_{y_i\in\mathcal{Y}_i} P(y_j \mid y_{\Pi_j})P(y_i \mid y_{\Pi_i})\\
&= \sum_{y_i\in\mathcal{Y}_i} p_{jy_j\pi_j}\,p_{iy_i\pi_i}.
\end{aligned}$$

The proof of the barren node removal easily follows from the fact that the vertex is not included anymore in the MID.

A.4 Proof of lemma 1 and 2

We first consider the arc reversal and the change of dimension of the vectors. If $j \notin \mathbb{J}$ the sets B_k that are affected by the arc reversal are only the ones such that $k \in \Pi_i \cup \Pi_j$ and the set B'_k simply takes into account the presence of the additional edges in G'. If $j \in \mathbb{J}'$ then the sets B_k affected by the arc reversal are the ones such that $k \in \Pi_i \cup \Pi_j \cup P_{j_j}$ and the set B''_k additionally takes into account that the indices in P_{j_j} are included only before the EUMarginalization between $\bar{U}_{i+1}$ and p_j. The final case is if $j \notin \mathbb{J}'$, which can be seen as a combination of the previous two cases.

Now consider the polynomial structure of the entries after an arc reversal. If $j \notin \mathbb{J}$, then the adjusted Algorithm 4.2 simply computes an EUMarginalization between $\bar{U}_{j+1}$ and p_i instead of p_j. Therefore the entries of $\bar{U}_j$ have $r'_{jba} = r_i r_{(j+1)ba}/r_j$ monomials of degree $d_{(j+1)ba}$ and, until the adjusted algorithm computes $\bar{U}_i$, the change in the structure is propagated through the 'EUOperations'. If $j \in \mathbb{J}' \cap \mathbb{J}$, then instead of an EUMultiSum and a EUMarginalization, now the algorithm only computes an EU-Marginalization and, as before, the change is propagated until $\bar{U}_i$. As in the previous paragraph, the last case can be seen as combination of the previous two situations.

Consider now the deletion of the barren node Y_i. The set B_z is the one with the highest index which includes i in G. Thus, for $i < k \leq z$, $i \in B_k$ and $\bar{U}_k$ is conditional on $Y_i = y_i$. The deletion of this vertex therefore implies that the dimension of the vector becomes c'_k/r_i. For $k \leq i$, Algorithm 4.2 now performs one EUMarginalization less and, from Proposition 4.2, we deduce that $\bar{U}'_k$ has now r_{kba}/r_i monomials of degree $d_{kba} - 1$.

A.5 Proof of proposition 6

Let $\Pi_{k\setminus i} = \Pi_k \setminus \{i\}$. If Y_i is parent of Y_k we have that

$$\begin{aligned} p'_{ky_k\pi'_k} &= P(y_k \mid y_{\Pi'_k}) = P(y_k \mid y_{\Pi_i}, y_{\Pi_{k\setminus i}}) = \sum_{y_i \in \mathcal{Y}_i} P(y_k \mid y_{\Pi_k}, y_{\Pi_i}) P(y_i \mid y_{\Pi_{k\setminus i}}, y_{\Pi_i}) \qquad (12) \\ &= \sum_{y_i \in \mathcal{Y}_i} P(y_k \mid y_{\Pi_k}) P(y_i \mid y_{\Pi_i}) = \sum_{y_i \in \mathcal{Y}_i} p_{ky_k\pi_k} p_{iy_i\pi_i} \end{aligned}$$

If Y_i is a parent but not the parent of Y_j, then $P(y_i \mid y_{\Pi_{j\setminus i}}, y_{\Pi_i})$ as in (12) can be written as

$$\begin{aligned} P(y_i \mid y_{\Pi_{j\setminus i}}, y_{\Pi_i}) &= P(y_i \mid y_{\Pi_j \setminus [i-1]}, y_{\Pi_j \cap [i-1]}, y_{\Pi_i}) \\ &= \frac{P(y_{\Pi_j \setminus [i-1]} \mid y_i, y_{\Pi_j \cap [i-1]}, y_{\Pi_i}) P(y_i \mid y_{\Pi_i})}{\sum_{y_i \in \mathcal{Y}_i} P(y_{\Pi_j \setminus [i-1]} \mid y_i, y_{\Pi_j \cap [i-1]}, y_{\Pi_i}) P(y_i \mid y_{\Pi_i})} \\ &= \frac{\prod_{l \in \Pi_j \setminus [i-1]} \sum_{\mathcal{Y}_{\Pi_i \cap \Pi_j \cap \Pi_l}} P(y_l \mid y_{\Pi_l}) P(y_i \mid y_{\Pi_i})}{\sum_{y_i \in \mathcal{Y}_i} \prod_{l \in \Pi_j \setminus [i-1]} \sum_{\mathcal{Y}_{\Pi_i \cap \Pi_j \cap \Pi_l}} P(y_l \mid y_{\Pi_l}) P(y_i \mid y_{\Pi_i})}, \end{aligned}$$

A.6 Proof of theorem 2

For $i, j, k, l \in \mathbb{V}$ and $s, t \in [m]$, an asymmetry $Y_i = y_i \Rightarrow Y_j = y_j$ implies that any monomials that include terms of the form $p_{ky_k\pi_k}$, $\psi_{s\pi_s}$, $p_{ky_k\pi_k} p_{ly_l\pi_l}$, $\psi_{t\pi_t}\psi_{s\pi_s}$ and $p_{ky_k\pi_k}\psi_{s\pi_s}$ entailing both instantiations y_i and y_j are associated to a non possible combination of events, with $y_k \in \mathcal{Y}_k$, $\pi_k \in \mathcal{Y}_{\Pi_k}$, $y_l \in \mathcal{Y}_l$, $\pi_l \in \mathcal{Y}_{\Pi_l}$, $\pi_t \in \mathcal{Y}_{P_t}$ and $\pi_s \in \mathcal{Y}_{P_s}$. Thus these monomials have to be set equal to zero.

For $j < t \leq z$, $\bar{U}_t$ has an associated set B_t which includes both i and j and consequently $\prod_{s \in B_t \setminus \{i \cup j\}} r_s$ rows of the vector corresponds to the conditioning on $Y_i = y_i$ and $Y_j = y_j$. Therefore all the monomials in those rows have to be set equal to zero.

For $i < t \leq j$, the index i is in the set B_t, whilst the variable Y_j has been already EUMarginalized. Thus, there are only $\prod_{s \in B_t \setminus \{i\}} r_s$ rows conditional on the event $Y_i = y_i$. In those rows only some of the monomials are associated to the event $Y_j = y_j$. Specifically, the ones implying $Y_j = y_j$ can only be multiplying a term including a ψ_{xP_x} from a utility vertex U_x subsequent to Y_j in the MID DS. We can deduce that there are $\prod_{s=t}^{j_a} r_s / r_j$ monomials of degree d_{tba} that include the case $Y_j = y_j$ in such entries of $\bar{U}_t$, for $a = x, \ldots, m$ and $b = l, \ldots, a$ (using the notation of Theorem 1).

Lastly, if $t \leq i$, then the set B_t does not include i and j, which have been both EUMarginalized. Thus monomials including a combination of the events $Y_j = y_j$ and $Y_i = y_i$ appears in each row of $\bar{U}_t$. Similarly as before, we can deduce that there are $\prod_{s=t}^{j_a} r_s / (r_i \cdot r_j)$ monomials of degree d_{tba}, $a = x, \ldots, m$, $b = l, \ldots, a$, implying the event $Y_i = y_i \wedge Y_j = y_j$.

Appendix B Maple code

B.1 Initialization functions

```
### Required Packages ###
with(ArrayTools): with(LinearAlgebra):

### Computation of the highest index in each parent set of a utility
node ###
# Inputs: PiU::table, parent sets of utility nodes; m::integer, num.
utility nodes
# Output: J::list

CompJ := proc(PiU,m) local i,j:
for j to m do J[j] := max(PiU[J]) end do:
return convert(J,list): end proc:

### Computation of the indices of the argument of the EU at step i ###
# Inputs: PiU::table; PiV::table, parent sets of random nodes;
i::integer;
# n::integer, number of random nodes; J::list
# Output: Bi[i]::set

CompBi := proc(PiU,PiV,i,n,J) local Bi,part,j:
Bi[i], part := {},{}:
for j from i to n do
part := part union {j}:
if member(j,V) then Bi[i] := Bi[i] union PiV[j] end if:
if member(j,J,'l') then Bi[i] := Bi[i] union PiU[l] end if:
end do:
Bi[i] := Bi[i] minus part:
return Bi[i]:
end proc:
```

```
### Initialization of an MID ###
# Inputs: p::table, probability vectors; psi::table, utility vectors;
PiV::table;
# PiU::table; n::integer; m::integer
# Outputs: J::list; Bi::list; u::table, EU vectors

Initialize := proc(p, psi, PiV, PiU, n, m) local J, i, Bi, u:
J := CompJ(PiU, m):
for i to n do Bi[i] := CompBi(PiU, PiV, i, n, J) end do:
Bi[n+1], u[n+1] := {}, []:
return J, Bi, u:
end proc:

### Identification of an optimal policy (at random) ###
# Inputs: r::table; i::integer, index of the decision variable,
# t::integer, number of random draws
#Outputs: maxi::vector, optimal decisions

Maximize := proc(r, i, t) local maxi, l:
maxi := Vector(t, 0):
for l to t do maxi[l] := RandomTools[Generate](integer(range = 1 ..
r[i])) end do:
return maxi:
end proc:
```

B.2 EU duplications

```
### EUDuplication of a utility vector and an EU vector ###
# Inputs: u::table; psi::table; j::integer; PiV::table; PiU::table;
# r::table, size of the decision and sample spaces; Bi::table; J::list
# Outputs: utemp::list, EUDuplicated version of u;
# psitemp::list, EUDuplicated version of psi

EUDuplicationPsi := proc(u, psi, j, PiV, PiU, r, Bi, J)
local i, uprime, psip, psit, utemp, x, sx, y, l, z:
i := max(PiU[j]):
uprime, psip, psit, utemp := [], [], psi[j], u[i+1]:
for x from max(Bi[i+1], PiU[j]) by -1 to 1 do
if member(x, (PiU[j] union Bi[i+1]) minus (PiU[j] intersect Bi[i+1]))
then sx := 1:
for y from x+1 to max(Bi[i+1], PiU[j]) do
if member(y, union(Bi[i+1], PiU[j])) then sx := sx*r[y] end if
end do:
if member(x, Bi[i+1]) then for l to Size(psit)[2]/sx do for z to r[x] do
psip := [op(psip),op(convert(convert(psit,list)[(l-1)*sx+1..l*sx],list))]
end do end do:
psit, psip := psip, []:
```

```
elif member(x, PiU[j]) then for l to Size(utemp)[2]/sx do for z to
r[x] do
uprime:=[op(uprime),op(convert(convert(utemp,list)[(l-1)*sx+1..l*sx],
list))] end do end do:
utemp, uprime := uprime, []:
end if end if end do:
return utemp, psit:
end proc:

### EUDuplication of a probability vector and an EU vector ###
# Inputs: u::table; p::table; i::integer; PiV::table; PiU::table;
r::table; Bi::table; J::list
# Outputs: utemp::list, EUDuplicated version of u;
# ptemp::list, EUDuplicated version of p

EUDuplicationP := proc (u, p, i, PiV, PiU, r, Bi, J)
local uprime, pprime, ptemp, utemp, x, sx, y, l, z, Uni:
uprime, pprime, ptemp, utemp := [], [], p[i], u[i+1]:
if member(i, J) then member(i, J, 'j');
Uni := (Bi[i+1] union PiV[i]) union PiU[j]:
for x from max(Uni) by -1 to 1 do
if member(x, Uni minus ((Bi[i+1] union PiU[j]) intersect (PiV[i]
union i))) then sx := 1;
for y from x+1 to max(Uni) do if member(y, Uni) then
sx := sx*r[y] end if end do;
if member(x, union(Bi[i+1], PiU[j])) then
for l to Size(ptemp)[2]/sx do for z to r[x] do
pprime:=[op(pprime),op(convert(convert(ptemp,Array)[(l-1)*sx+1..l*sx],
list))]
end do end do:
ptemp, pprime := pprime, []:
elif member(x, PiV[i]) then for l to Size(utemp)[2]/sx do
for z to r[x] do uprime:=[op(uprime),op(convert(convert(utemp,Array)
[(l-1)*sx+1..l*sx],list))]
end do end do:
utemp, uprime := uprime, []:
end if end if end do:
else for x from max(Bi[i+1], PiV[i]) by -1 to 1 do
if member(x,(Bi[i+1] union PiV[i])minus(Bi[i+1] intersect (PiV[i]
union i))) then sx := 1;
for y from x+1 to max(Bi[i+1],PiV[i]) do if member(y,Bi[i+1] union
PiV[i]) then sx := sx*r[y]
end if end do:
if member(x, Bi[i+1]) then for l to Size(ptemp)[2]/sx do for z to
r[x] do
pprime:=[op(pprime),op(convert(convert(ptemp,Array)[(l-1)*sx+1..l*sx],
list))]
end do end do:
```

```
ptemp, pprime := pprime, []:
elif member(x, PiV[i]) then for l to Size(utemp)[2]/sx do for z to
r[x] do
uprime:=[op(uprime),op(convert(convert(utemp,Array)[(l-1)*sx+1..l*sx],
list))]
end do end do;
utemp, uprime := uprime, []:
end if end if end do end if:
utemp, ptemp := convert(utemp,Array), convert(ptemp,Array):
return utemp,ptemp: end proc:
```

B.3 EU operations

```
### EuMultiSum between an EU vector and a utility vector ###
# Inputs: u::table; psi::table; j::integer; PiV::table; PiU::table;
# r::table; Bi::table; J::list
# Outputs: ut::list, EU vector after an EUMultiSum

EUMultiSum := proc(u, psi, j, PiV, PiU, r, Bi, J) local i, uprime,
psip, ut; i := max(PiU[j]);
if j = Size(convert(PiU, list), 2) then ut := k[j]* psi[j]:
else uprime, psip := EUDuplicationPsi(u, psi, j, PiV, PiU, r, Bi, J);
ut := h* k[j]* psip* uprime +  uprime +  k[j]* psip end if:
return ut:
end proc:

### EUMarginalization over a sample space ###
# Inputs: u::table; p::table; i::integer; PiV::table; PiU::table;
r::table; Bi::table; J::list
# Outputs: ut::list, EU vector after EUMarginalization

EUMarginalization := proc (u, p, i, PiV, PiU, r, Bi, J)
local uprime, pprime, ut, cols, l, k:
uprime, pprime := EUDuplicationP(u, p, i, PiV, PiU, r, Bi, J): cols
:= Size(pprime)[2]:
ut := convert(ZeroVector(cols/r[i]), Array):
for l to (cols/r[i]) do for k to r[i] do ut[l] := ut[l]+
pprime[r[i]*(l-1)+k]*uprime[r[i]*(l-1)+k]:
end do end do:
return ut:
end proc:

### EUMaximization over a decision space ###
# Inputs: u::table; i::integer; r::table
# Outputs: u[i]::list, EU vector after
EUMaximization
```

```
EUMaximization := proc(u, i, r) local opt,l ;
opt := Maximize(r, i, Size(u[i+1])[2]/r[i]);
u[i] := Array([seq(0,l in 1..Size(opt)[1])]):
for l to Size(opt)[1] do
u[i][l] := convert(u[i+1],Array)[r[i]*(l-1)+opt[l]] end do;
return u[i]:
end proc:
```

B.4 The symbolic algorithm

```
### Symbolic evaluation algorithm for an MID ###
# Inputs: p::table; psi::table; PiV::table; PiU::table; n::integer;
m::integer;
# De::set, index set of the decision variables;
# V::set, index set of the random variables; r::table
# Output: eu::table, EU vectors;

SymbolicExpectedUtility := proc(p, psi, PiV, PiU, n, m, De, V, r)
local J, Bi, utemp, i, j, eu;
J, Bi, eu := Initialize(p, psi, PiV, PiU, n, m);
j := m;
for i from n by -1 to 1 do if j = 0 then if member(i, De) then eu[i]
:= EUMaximization(eu, i, r)
else eu[i] := EUMarginalization(eu, p, i, PiV, PiU, r, Bi, J) end if;
else if J[j] = i then if member(i, De) then
utemp[i+1] := EUMultiSum(eu, psi, j, PiV, PiU, r, Bi, J);
eu[i] := EUMaximization(utemp, i, r)
else
utemp[i+1] := EUMultiSum(eu, psi, j, PiV, PiU, r, Bi, J);
eu[i] := EUMarginalization(utemp, p, i, PiV, PiU, r, Bi, J)
end if;
j := j-1
else if member(i, De) then eu[i] := EUMaximization(eu, i, r)
else eu[i] := EUMarginalization(eu, p, i, PiV, PiU, r, Bi, J) end if
end if end if end do;
return eu:
end proc:
```

B.5 Implementation of the example

Consider the MID in Fig. 1 with $n = 6$ variables (decision or random nodes) and $m = 3$ utility nodes.

```
### Definition of the MID ###
# number of variables and utility nodes
n := 6: m := 3:
# V contains the indices of random nodes and De those of the decision
nodes V := 2, 3, 5, 6: De := 1, 4:
# Conditional probabilities
```

```
p[6] := [p6111, p6011, p6101, p6001, p6110, p6010, p6100, p6000]:
p[5] := [p5111, p5011, p5101, p5001, p5110, p5010, p5100, p5000]:
p[3] := [p3111, p3011, p3101, p3001, p3110, p3010, p3100, p3000]:
p[2] := [p211, p201, p210, p200]:
# Utility parameters
psi[1] := [psi11, psi10]:
psi[2] := [psi21, psi20]:
psi[3] := [psi311, psi301, psi310, psi300]:
# Parents of random nodes
PiV[2] := 1: PiV[3] := 1, 2: PiV[5] := 3, 4: PiV[6] := 4, 5:
# Parents of utility nodes
PiU[1] := 3: PiU[2] := 5: PiU[3] := 4, 6:
# Number of levels of the variables
r[1] := 2: r[2] := 2: r[3] := 2: r[4] := 2: r[5] := 2: r[6] := 2:
### Computation of the EU vectors ###
eu := SymbolicExpectedUtility(p, psi, PiV, PiU, n, m, De, V, r):
```

Example of the output of eu[1]:

```
[((k[1]*psi11+h*k[1]*psi11*((k[2]*psi21+h*k[2]*psi21*
(p6010*psi300*k[3]+p6110*psi310*k[3])+k[3]*psi300*p6010
+k[3]*psi310*p6110)*p5101+(k[2]*psi20+h*k[2]*psi20*(p6000*psi300*k[3]
+p6100*psi310*k[3])+k[3]*psi300*p6000+k[3]*psi310*p6100)*p5001)+
(k[2]*psi21+h*k[2]*psi21*(p6010*psi300*k[3]+p6110*psi310*k[3])
+k[3]*psi300*p6010+k[3]*psi310*p6110)*p5101+
(k[2]*psi20+h*k[2]*psi20*(p6000*psi300*k[3]+p6100*psi310*k[3])
+k[3]*psi300*p6000+k[3]*psi310*p6100)*p5001)*p3110+
(k[1]*psi10+h*k[1]*psi10*((k[2]*psi21+h*k[2]*psi21*(p6011*psi301*k[3]
+p6111*psi311*k[3])+k[3]*psi301*p6011+k[3]*psi311*p6111)*p5110+
(k[2]*psi20+h*k[2]*psi20*(p6001*psi301*k[3]+p6101*psi311*k[3])
+k[3]*psi301*p6001+k[3]*psi311*p6101)*p5010)+
(k[2]*psi21+h*k[2]*psi21*(p6011*psi301*k[3]+p6111*psi311*k[3])
+k[3]*psi301*p6011+k[3]*psi311*p6111)*p5110+
(k[2]*psi20+h*k[2]*psi20*(p6001*psi301*k[3]+p6101*psi311*k[3])
+k[3]*psi301*p6001+k[3]*psi311*p6101)*p5010)*p3010)*p210+
((k[1]*psi11+h*k[1]*psi11*((k[2]*psi21+h*k[2]*psi21*(p6010*psi300*k[3]
+p6110*psi310*k[3])+k[3]*psi300*p6010+k[3]*psi310*p6110)*p5101+
(k[2]*psi20+h*k[2]*psi20*(p6000*psi300*k[3]+p6100*psi310*k[3])
+k[3]*psi300*p6000+k[3]*psi310*p6100)*p5001)+
(k[2]*psi21+h*k[2]*psi21*(p6010*psi300*k[3]+p6110*psi310*k[3])
+k[3]*psi300*p6010+k[3]*psi310*p6110)*p5101+
(k[2]*psi20+h*k[2]*psi20*(p6000*psi300*k[3]+p6100*psi310*k[3])
+k[3]*psi300*p6000+k[3]*psi310*p6100)*p5001)*p3100+
(k[1]*psi10+h*k[1]*psi10*((k[2]*psi21+h*k[2]*psi21*
(p6011*psi301*k[3]+p6111*psi311*k[3])+k[3]*psi301*p6011+k[3]
psi311*p6111)*p5110+ (k[2]*psi20+h*k[2]*psi20*(p6001*psi301*k[3]+
p6101*psi311*k[3]) +k[3]*psi301*p6001+k[3]*psi311*p6101)*p5010)+
(k[2]*psi21+h*k[2]*psi21*(p6011*psi301*k[3]+p6111*psi311*k[3])
+k[3]*psi301*p6011+k[3]*psi311*p6111)*p5110+
(k[2]*psi20+h*k[2]*psi20*(p6001*psi301*k[3]+p6101*psi311*k[3])
+k[3]*psi301*p6001+k[3]*psi311*p6101)*p5010)*p3000)*p200]
```

References

1. Bhattacharjya, D., Shachter, R.D.: Sensitivity Analysis in Decision Circuits. In: Proceedings of the 24th Conf. Uncertainty in Artif. Intel, pp. 34–42 (2008)
2. Bhattacharjya, D., Shachter, R.D.: Three New Sensitivity Analysis Methods for Influence Diagrams. In: Proc. 26Th Conf. Uncertainty in Artif. Intel., pp. 56–64 (2010)
3. Bhattacharjya, D., Shachter, R.D.: Formulating asymmetric decision problems as decision circuits. Decis. Anal. **9**, 138–145 (2012)
4. Bielza, C., Gómez, M., Shenoy, P.P.: A review of representation issues and modeling challenges with influence diagrams. Omega **39**, 227–241 (2011)
5. Blekherman, G., Parrilo, P.A., Thomas, R.R.: Semidefinite optimization and convex algebraic geometry. Siam philadelphia (2013)
6. Borgonovo, E., Tonoli, F.: Decision-network polynomials and the sensitivity of decision-support models. Eur. J. Oper. Res. **239**, 490–503 (2014)
7. de Campos, C.P., Cozman, F.G.: Inference in Credal Networks Using Multilinear Programming. In: Proceedings of the 2Nd Starting AI Researcher Symp., pp. 50–61 (2004)
8. de Campos, C.P., Ji, Q.: Strategy Selection in Influence Diagrams Using Imprecise Probabilities. In: Proceedings of the 24Th Conf. Uncertainty in Artif. Intel., pp. 121–128 (2008)
9. Castillo, E., Gutierrez, J.M., Hadi, A.S.: Parametric Structure of Probabilities in Bayesian Networks. In: ECSQARU 1995, pp. 89–98. Springer (1995)
10. Castillo, E., Gutierrez, J.M., Hadi, A.S.: Sensitivity analysis in discrete Bayesian networks. IEEE T. Syst. Man. Cy. A **27**, 412–423 (1997)
11. Castillo, E., Gutierrez, J.M., Hadi, A.S.: Expert Systems and Probabilistic Network Models. Springer, New York (2012)
12. Castillo, E., Kjærulff, U.: Sensitivity analysis in Gaussian Bayesian networks using a symbolic-numerical technique. Reliab. Eng. Syst. Safe. **79**, 139–148 (2003)
13. Chan, H., Darwiche, A.: When Do Numbers Really Matter? In: Proceedings of the 17th Conf. Uncertainty in Artif. Intel., pp. 65–74 (2001)
14. Chan, H., Darwiche, A.: Sensitivity Analysis in Bayesian Networks: from Single to Multiple Parameters. In: Proceedings of the 20Th Conf. Uncertainty in Artif. Intel., pp. 67–75 (2004)
15. Coupé, V.M.H., van der Gaag, L.C.: Properties of sensitivity analysis of Bayesian belief networks. Ann. Math. Artif. Intel. **36**, 323–356 (2002)
16. Cox, D.A., Little, J., O'Shea, D.: Ideals, Varieties, and Algorithms. Springer, New York (2007)
17. Cozman, F.G.: Credal networks. Artif. Intel. **120**, 199–233 (2000)
18. Darwiche, A.: A differential approach to inference in Bayesian networks. J. ACM **50**, 280–305 (2003)
19. Dawid, A.P., Constantinou, P.: A formal treatment of sequential ignorability. Stat. Biosci. **6**, 166–188 (2014)
20. Demirer, R., Shenoy, P.P.: Sequential valuation networks for asymmetric decision problems. Eur. J. Oper. Res. **169**, 286–309 (2006)
21. Felli, J.C., Hazen, G.B.: Javelin diagrams: a graphical tool for probabilistic sensitivity analysis. Decis. Anal. (2) (2004)
22. French, S.: Readings in Decision Analysis. CRC Press, Boca Raton (1989)
23. van der Gaag, L.C., Renooij, S., Coupé, V.M.H.: Sensitivity Analysis of Probabilistic Networks. In: Advances in Probabilistic Graphical Models, pp. 103–124. Springer (2007)
24. Görgen, C., Leonelli, M., Smith, J.Q.: A Differential Approach for Staged Trees. In: ECSQARU 2015, pp. 346–355. Springer (2015)
25. Howard, R.: The foundations of decision analysis. IEEE T. Syst. Man. Cyb. **4**, 211–219 (1968)
26. Howard, R.A., Matheson, J.E.: Influence diagrams. Decis. Anal. **2**, 127–143 (2005)

27. Jensen, F., Jensen, F.V., Dittmer, S.L.: From Influence Diagrams to Junction Trees. In: Proceedings 10th Conf. Uncertainty in Artif. Intel., pp. 367–373 (1994)
28. Jensen, F.V., Nielsen, T.D., Shenoy, P.P.: Sequential influence diagrams: a unified asymmetry framework. Int. J. Approx. Reason. **42**, 101–118 (2006)
29. Keeney, R.L.: Multiplicative utility functions. Oper. Res. **22**, 22–34 (1974)
30. Keeney, R.L., Raiffa, H.: Decision with Multiple Objectives. Cambridge University Press, Cambridge (1976)
31. Kikuti, D., Cozman, F.G., Shirota Filho, R.: Sequential decision making with partially ordered preferences. Artif. Intel. **175**, 1346–1365 (2011)
32. Koller, D., Friedman, N.: Probabilistic Graphical Models. MIT press, Cambridge (2009)
33. Leonelli, M., Görgen, C., Smith, J.Q.: Sensitivity analysis, multilinearity and beyond. Tech. rep., arXiv:1512.02266 (2015)
34. Leonelli, M., Smith, J.Q.: Bayesian decision support for complex systems with many distributed experts. Ann. Oper. Res. **235**, 517–542 (2015)
35. Nielsen, T.D., Jensen, F.V.: Sensitivity analysis in influence diagrams. IEEE T. Syst. Man. Cy. A **33**, 223–234 (2003)
36. Nielsen, T.D., Jensen, F.V.: Bayesian Networks and Decision Graphs. Springer, New York (2009)
37. Pearl, J.: Probabilistic Reasoning in Intelligent Systems. Morgan Kaufmann, San Francisco (1988)
38. Sanner, S., Kersting, K.: Symbolic Dynamic Programming for First-Order POMDPs. In: Proceedings 24Th AAAI Conf. Artif. Intel., pp. 1140–1146 (2010)
39. Shachter, R.D.: Evaluating influence diagrams. Oper. Res. (6) (1986)
40. Shachter, R.D.: An ordered examination of influence diagrams. Networks (20) (1990)
41. Smith, J.E., Holtzman, S., Matheson, J.E.: Structuring conditional relationships in influence diagrams. Oper. Res. **41**, 280–297 (1993)
42. Smith, J.Q.: Influence diagrams for Bayesian decision analysis. Eur. J. Oper. Res. **40**, 363–376 (1989)
43. Smith, J.Q.: Influence diagrams for statistical modelling. Ann. Stat. **17**, 654–672 (1989)
44. Smith, J.Q.: Bayesian Decision Analysis: Principles and Practice. Cambridge University Press, Cambridge (2010)
45. Smith, J.Q., Barons, M.J., Leonelli, M.: Coherent frameworks for statistical inference serving integrating decision support systems. Tech. rep., CRISM15-10, Warwick University (2015)
46. Tatman, J.A., Shachter, R.D.: Dynamic programming and influence diagrams. IEEE Trans. Systems Man Cybernet. **20**, 365–379 (1990)
47. Zamani, Z., Sanner, S., Fang, C.: Symbolic Dynamic Programming for Continuous State and Action MDPs. In: Proceedings 26th AAAI Conf. Artif. Intel., pp. 1839–1845 (2012)