Using computer algebra to symbolically evaluate discrete influence diagrams

Manuele Leonelli[1], James Q. Smith

Department of Statistics, The University of Warwick, CV47AL Coventry, UK

Eva Riccomagno

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

Abstract

Influence diagrams provide a compact graphical representation of decision problems. Several algorithms for the quick computation of their associated expected utilities have been developed. However, these often rely on a full quantification of the uncertainties and values required for the calculation of these expected utilities. Here we develop a symbolic way to evaluate influence diagrams, not requiring such a full numerical specification, for the case when random variables and decisions are all discrete. Within this approach expected utilities correspond to families of polynomials. This polynomial representation enables us to study many important features of an influence diagram. First, we develop an efficient symbolic algorithm for the propagation of expected utilities through the diagram and we provide an implementation of this algorithm using a computer algebra system. Second, we characterize many of the standard manipulations of these diagrams as transformations of polynomials. Third, we identify classes of influence diagrams which are equivalent in the sense that they all share the same polynomial structure. Finally, we generalize the decision analytic framework of the influence diagram by characterizing asymmetries as manipulations of the expected utility polynomials.

Keywords: Asymmetric Decision Problems, Computer Algebra, Influence Diagrams, Lattices, Symbolic Inference.

1. Introduction

Decision makers (DMs) are often required in critical situations to choose 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. Empirical evidence has shown that, due to this complexity, DMs often struggle to act as rational expected utility maximizers and instead tend to exhibit inconsistencies in their behaviour [1]. They can therefore benefit from an intuitive framework which draws together these different uncertainties and values so as to better understand and evaluate the full consequences of the assumptions they are making.

Influence diagrams (IDs) [2, 3] provide such a platform to frame and review the elements of a DM's decision problem. This has been successfully applied in a variety of domains (see for example [4, 5] for a review). For example [6] catalogues over 250 documented practical application of IDs models. IDs are directed acyclic graphs (DAGs) that can be thought of as Bayesian network (BN) augmented with decision and value nodes. IDs not only provide a compact graphical representation of both the structure of a decision problem and the relationships between its relevant uncertainties, but also offer a computational framework for the fast evaluation of the expected utility scores of different policies. Several algorithms have now been

[1]Corresponding author. E-mail: *m.leonelli@warwick.ac.uk*, Phone : 00447593389567, Address: Department of Statistics, The University of Warwick, CV47AL Coventry, UK.

Preprint submitted to International Journal of Approximate Reasoning *May 12, 2015*

defined to evaluate influence diagrams, i.e. identify an expected utility maximizing policy and its associated score. These can work directly on the ID [7, 8], transform it into a decision tree [9], or convert the diagram into some other graphical structure [10, 11]. There were severe computational issues related to the evaluation of IDs, but there are now several promising methodologies designed to overcome these challenges as noted in [4] and [12]. Many different pieces of software are also now available to build and automatically evaluate large IDs [5].

Most of the above mentioned evaluation algorithms work numerically once a *full* probability and utility elicitation is available to the DM. However, often in practice DMs are not confident about the precision of these values. They also may simply not be ready to provide such quantifications because of time or economic constraints. Here we address these challenges by developing a symbolic approach to the evaluation of IDs which does not require a full elicitation of all the quantities needed to compute expected scores. The symbolic definition of the ID's probabilities and utilities provides an elegant and efficient embellishment of the associated graphical representation of the problem, around which symbolic computations can then be carried. This symbolic framework is especially useful for *sensitivity analyses* [13, 14, 15, 16], where, for example, DMs can simply plug in different estimates of the parameters of their problem and observe how these changes influence the outcome of the evaluation.

For simplicity, as in other discussions [17, 18, 19, 20] concerning BNs and decision circuits, we assume in this paper that random variables and decisions are all discrete. Generalizing the work in [21], we make heavy use of the fact that, within our symbolic approach, the expected utility of a specific policy is a polynomial, whose unknown quantities are conditional probabilities, marginal utilities and criterion weights. We develop a symbolic algorithm for the computation of this polynomial representation of an ID model, which is specified through simple matrix operations. Because of the simplicity of its operations, our implementation in the MapleTH[2] computer algebra system (reported in Appendix B) is able to compute these polynomials instantaneously in our examples once the symbolic definition of the ID is given as input. In contrast to standard software, which assumes an additive factorization between utility nodes, our code is able to explicitly model multiplicative utility factorizations [22].

In larger real-world applications computer algebra systems can have difficulties handling the number of unknown variables that need to be stored in the computer memory. When this happens computations often become infeasible. However, by imposing certain conditions on the model - formally discussed in [23] - computations can then be distributed in such a way that the ID represents solely the input/output relationship between large-dimensional models. This can dramatically reduce complexity. A random node of the ID will then correspond to the relevant outputs of a large statistical model and arrows connecting any two nodes would imply that the outputs of one model are used as input for the other. When such a representation is possible, computer algebra can then be employed even when addressing much larger applications.

The new symbolic representation of decision problems we introduce here allows us to concisely express a large amount of information that might not be apparent from an ID, with a potential for making computations much more efficient. Different types of extra information, often consisting of asymmetries of various kinds, have been explicitly modelled in graphical extensions of the ID model [24, 25, 26, 27, 28, 29, 30]. A significant proportion of these, although providing a framework for the evaluation of more general decision problems, 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 expected utility.

The structure of the paper is as follows. Section 2 introduces a new class of IDs entertaining multiplicative utility factorizations and discusses the polynomial structure of the associated expected utilities. Section 3 presents our new symbolic algorithm and Section 4 symbolically defines manipulations of the graphical representation of an ID. In Section 5 we discover a lattice structure between IDs describing the same decision problem, which can then be of use in sensitivity analyses and model choice. Section 6 deals with asymmetries and formalizes their symbolic interpretation. We conclude with a discussion.

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

2. Representation of influence diagrams

In this section we assume the DM deals with a uniform (or symmetric) decision problem, consisting of the class of problems that can be represented by an ID [31, 32]. We consider asymmetric decision problems in Section 6. From the definition of the ID we deduce the structure of its conditional expected utility functions and show that these are explicit polynomial functions whose arguments are derived from either the random nodes or the utility nodes of the ID.

Let $[n] = \{1, \dots, n\}$ and $\{Y_1, \dots, Y_n\}$ be a partially ordered set according to an order $\preceq$ such that if $Y_i \preceq Y_j$, then $i < j$, for any $i, j \in [n]$. We assume that $[n]$ is partitioned into $\mathbb{D}$ and $\mathbb{V}$, both non-empty sets. Also $\{Y_1, \dots, Y_n\}$ results to be partitioned in $\{Y_i : i \in \mathbb{D}\}$ and $\{Y_i : i \in \mathbb{D}\}$, called the sets of controlled (or decision)[3] and non-controlled (or random) variables, respectively. Furthermore we assume $\{Y_i : i \in \mathbb{D}\}$ to be totally ordered: this is an intrinsic requirement of any symmetric decision problem. The partial order $\preceq$ satisfying the total order on $\{Y_i : i \in \mathbb{D}\}$, can be arbitrarily refined to a total order and the choice of the total order does not affect the theory presented here.

For $i \in [n]$, we assume that Y_i takes values in $\mathcal{Y}_i = \{0, \dots, r_i - 1\}$, where $r_i \in \mathbb{Z}_{\geq 1}$. For $A \subseteq [n]$, $\boldsymbol{Y}_A = (Y_i)^{\mathrm{T}}_{i \in A}$ takes values in $\boldsymbol{\mathcal{Y}}_A = \times_{i \in A} \mathcal{Y}_i$ and $\boldsymbol{Y}^i_A = (Y_j)^{\mathrm{T}}_{j \in A, j < i}$. We denote with $\boldsymbol{y}_A$ and $\boldsymbol{y}^i_A$ generic instantiations of $\boldsymbol{Y}_A$ and $\boldsymbol{Y}^i_A$ respectively.

Example 1. A few important examples of the notation above are: the vector $\boldsymbol{Y}_{[n]}$ includes all the variables in $\{Y_1, \dots, Y_n\}$, $\boldsymbol{Y}_{\mathbb{D}}$ and $\boldsymbol{Y}_{\mathbb{V}}$ are the vectors of controlled and random variables respectively, and $\boldsymbol{Y}^{n-1}_{[n]} = (Y_1, \dots, Y_{n-2})^{\mathrm{T}}$.

2.1. Multiplicative influence diagram

We consider here a class of IDs entertaining a factorization over its utility nodes known as *multiplicative* [22, 32]. We consider an m-dimensional vector of utilities $\boldsymbol{U} = (U_1, \dots, U_m)^{\mathrm{T}}$ and for each $i \in [m]$ we consider $P_i \subseteq [n]$ non empty (the elements of $\boldsymbol{Y}_{P_i}$ are the parents of U_i, as we see next). The i-th component of $\boldsymbol{U}$, U_i, is a utility function mapping a subspace $\boldsymbol{\mathcal{Y}}_{P_i}$ of $\boldsymbol{\mathcal{Y}}_{[n]}$ into $[0, 1]$.

Definition 1. A multiplicative influence diagram (MID) G consists of

- a DAG with vertex (or node) set $V(G) = \{Y_1, \dots, Y_n, U_1, \dots, U_m\}$ and edge (or arc) set $E(G)$ including three types of edges:
  1. for $i \in [m]$, U_i has no children, its parent set is given by the elements of $\boldsymbol{Y}_{P_i}$ and an element of $\boldsymbol{Y}_{[n]}$ is parent of at most one element of $\boldsymbol{U}$. 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 $\{Y_1, \dots, Y_n\}$: formally, if there is a $k \in P_i$ such that for every $l \in P_j$, $k > l$;
  2. for $i \in \mathbb{D}$, the parent set of Y_i, $\{Y_j : j \in \Pi_i \subset [i-1]\}$, consists of the variables, controlled and non-controlled, that are known when Y_i is controlled;
  3. for $i \in \mathbb{V}$, the parent set of Y_i, $\{Y_j : j \in \Pi_i \subset [i-1]\}$ is such that $Y_i \perp\!\!\!\perp \boldsymbol{Y}^i_{\mathbb{V} \setminus \Pi^{\mathbb{V}}_i} \mid \boldsymbol{Y}_{\Pi^{\mathbb{V}}_i}$ for all instantiations of decisions preceding Y_i. Here $\Pi^{\mathbb{V}}_i = \Pi_i \cap \mathbb{V}$ and conditional independence is with respect to the probability law defined next;[4]
- for $i \in \mathbb{V}$, a transition density function $P(y_i \mid \boldsymbol{y}_{\Pi_i}) = P(Y_i = y_i \mid \boldsymbol{Y}_{\Pi_i} = \boldsymbol{y}_{\Pi_i})$;[5]

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

[4]More succinctly this requirement can be written as $Y_i \perp\!\!\!\perp \boldsymbol{Y}^i_{[n]} \mid \boldsymbol{Y}_{\Pi_i}$ using the notion of generalized conditional independence recently introduced in [33].

[5]Note that we allow transition densities to be conditional on controlled variables.

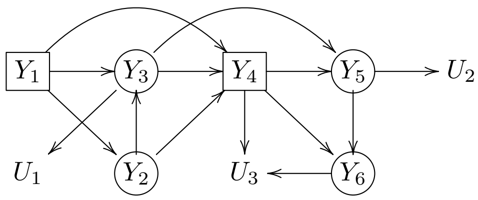


Figure 1: An ID describing the available countermeasures after an accident at a nuclear power plant.

- a multiplicative factorization over U such that

$$U(\boldsymbol{y}_{[n]}) = \sum_{I \in \mathcal{P}_0([m])} h^{n_I - 1} \prod_{i \in I} k_i U_i(\boldsymbol{y}_{P_i}), \tag{1}$$

  where $\mathcal{P}_0(\cdot)$ denotes the power set without the empty set, n_I is the number of elements in the set I, $k_i \in (0,1)$ is a **criterion weight** [32], U_i is a function of $\boldsymbol{Y}_{P_i}$ only and h is the unique solution not smaller than minus one to

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

We concentrate our study on the popular class of multiplicative factorizations because this provides some computational advantages over, for example, the multilinear one [22], whilst allowing for enough flexibility to model the DM's preferences in many real cases [34]. Also the famous class of additive factorizations [22, 35] can be seen as special cases of the multiplicative class for $h = 0$, assuming $0^0 = 1$. Thereof an ID whose associated decision problem implies an additive utility factorization is called additive ID.

Item 1 in Definition 1 extends the total order over $\{Y_1, \dots, Y_n\}$ to $V(G)$. For $i \in [m]$, let j_i be the highest index of P_i and $\mathbb{J} = \{j_1, \dots, j_m\}$. The MapleTH function `CompJ` in Appendix B receives as input the number m and the parent set of U_i, for all $i \in [m]$, and 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 indicated as $S := (Y_1, \dots, Y_{j_1}, U_1, Y_{j_1+1}, \dots, Y_{j_m}, U_m)$. Differently from other authors [e.g. 31, 36], 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 2. Figure 1 presents an MID with vertex set $\{Y_1, \dots, Y_6, U_1, \dots, U_3\}$ describing a gross simplification of the possible countermeasures after an accident at a nuclear power plant. For this MID, $n = 6$, $m = 3$, $\mathbb{D} = \{1, 4\}$ and $\mathbb{V} = \{2, 3, 5, 6\}$. Therefore, there are two controlled variables Y_1 and Y_4: the first consisting of the possibility of closing down the power plant, the second of evacuating the nearby population. Before controlling Y_4, the variables Y_1, Y_2 and Y_3 are observed since $\Pi_4 = [3]$. We adopt the convention by which decision variables and random variables are respectively framed with squares and circles. There are four random nodes: Y_2 and Y_3 measure the amount of dispersion of the contaminant in the atmosphere and to the ground respectively, Y_5 estimates the human intake of radiation and Y_6 ranks the level of stress in the population. This DAG implies that $Y_5 \perp\!\!\!\perp Y_2 \mid Y_3$ for every $y_1 \in \mathcal{Y}_1$ and $y_4 \in \mathcal{Y}_4$. The variable Y_5 has parent set $\{Y_i : i \in \Pi_5\}$ where $\Pi_5 = \{3, 4\}$ and $\Pi_5^{\mathbb{V}} = \{3\}$. All variables are binary and take values in the spaces $\mathcal{Y}_i = \{1, 0\}$, $i \in [6]$. When $\mathcal{Y}_i$ is associated to a decision node, 1 and 0 correspond to, respectively, proceed and not proceed. If $\mathcal{Y}_i$ is associated to a random node, then 1 and 0 correspond respectively to high and low. The MID in Figure 1 is completed by three utility nodes, U_1, U_2 and U_3. The set P_3 is equal to $\{4, 6\}$ and therefore Y_4 and Y_6 are the arguments of the utility function U_3. Lastly, the DS associated to the MID is $(Y_1, Y_2, Y_3, U_1, Y_4, Y_5, U_2, Y_6, U_3)$ and $j_1 = 3$, $j_2 = 5$, $j_3 = 6$: thus $\mathbb{J} = \{3, 5, 6\}$.

2.2. The evaluation of multiplicative influence diagrams

To *evaluate* an MID is to identify a sequence of optimal decisions maximizing the expected utility function (i.e. the expectation of equation (1)). However, only MIDs whose topology is such that, for any index $j \in \mathbb{D}$, the only variables that are known at the time the DM makes a decision Y_j have an index lower than j can be directly evaluated using 'extensive form'standard calculations [37]. This is because the evaluation will output optimal decisions as functions of observed quantities only [32].

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 $i < j$.

We mainly study MIDs in extensive form and in Section 4 consider manipulations of non extensive MIDs. Without loss of generality we assume that every vertex Y_i of the MID has at least one child. Random and controlled vertices with no children could be simply deleted from the graph without changing the outcome of the evaluation [see e.g. 31].

Example 3. The MID in Figure 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. The only vertices with no children are utility nodes.

A simple way to evaluate an MID in extensive form is through a backward inductive algorithm over the vertices of the DAG [31]. Here we introduce a computationally efficient new version of this algorithm, including at each step only the strictly necessary utility nodes. The identification of the optimal policy is based on the computation of the functions $\bar{U}_i(\boldsymbol{y}_{B_i})$, $i \in [n]$, we formally introduce in Proposition 1, where

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

is the set including the indices of the variables that appear as arguments of $\bar{\boldsymbol{U}}_i$. It should be noted that B_i includes only indices smaller than i that are either in the parent set of a random variable Y_k, $k > i$, or in a set P_j such that u_j succeeds Y_i in the DS of the MID.

Example 4. We compute here the sets B_5 and B_4 associated with the MID in Figure 1. The set $B_5 = \{3, 4\}$ since $B_5 = \{\Pi_6 \cup \Pi_5 \cup P_3 \cup P_2\} \setminus \{5, 6\}$, where $\Pi_6 = \{4, 5\}$, $\Pi_5 = \{3, 4\}$, $P_3 = \{4, 6\}$ and $P_2 = \{5\}$. Furthermore $B_4 = \{3\}$ since it can be noted that $B_4 = B_5 \setminus \{4\}$.

Proposition 1. *The optimal decision associated to an MID yields expected utility* $\bar{U}_1(\boldsymbol{y}_{B_1})$*, where, for* $i \in [n]$*,* $\bar{U}_i(\boldsymbol{y}_{B_i})$ *is defined as*

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

where, for $i = n$*,*

$$\bar{U}_n^{\mathbb{D}}(\boldsymbol{y}_{B_n}) = \max_{\mathcal{Y}_n} k_m U_m(\boldsymbol{y}_{P_m}), \qquad \bar{U}_n^{\mathbb{V}}(\boldsymbol{y}_{B_n}) = \int_{\mathcal{Y}_n} k_m U_m(\boldsymbol{y}_{P_m}) f_n(y_n \mid \boldsymbol{y}_{\Pi_n}) \mathrm{d}y_n, \tag{3}$$

and, for $i \in [n-1]$*, if* $i \in \mathbb{J}$ *and supposing* $i \in P_l$*,*

$$\bar{U}_i^{\mathbb{D}}(\boldsymbol{y}_{B_i}) = \max_{\mathcal{Y}_i} \Big(h k_l U_l(\boldsymbol{y}_{P_l}) \bar{U}_{i+1}(\boldsymbol{y}_{B_{i+1}}) + k_l U_l(\boldsymbol{y}_{P_l}) + \bar{U}_{i+1}(\boldsymbol{y}_{B_{i+1}}) \Big), \tag{4}$$

$$\bar{U}_i^{\mathbb{V}}(\boldsymbol{y}_{B_i}) = \int_{\mathcal{Y}_i} \Big(h k_l U_l(\boldsymbol{y}_{P_l}) \bar{U}_{i+1}(\boldsymbol{y}_{B_{i+1}}) + k_l U_l(\boldsymbol{y}_{P_l}) + \bar{U}_{i+1}(\boldsymbol{y}_{B_{i+1}}) \Big) f_i(y_i \mid \boldsymbol{y}_{\Pi_i}) \mathrm{d}y_i, \tag{5}$$

whilst, if $i \notin \mathbb{J}$*,*

$$\bar{U}_i^{\mathbb{D}}(\boldsymbol{y}_{B_i}) = \max_{\mathcal{Y}_i} \bar{U}_{i+1}(\boldsymbol{y}_{B_{i+1}}), \qquad \bar{U}_i^{\mathbb{V}}(\boldsymbol{y}_{B_i}) = \int_{\mathcal{Y}_i} \bar{U}_{i+1}(\boldsymbol{y}_{B_{i+1}}) f_i(y_i \mid \boldsymbol{y}_{\Pi_i}) \mathrm{d}y_i. \tag{6}$$

The proof of this theorem is in Appendix A.1. Equation (3) consists of a marginalization/maximization over $\mathcal{Y}_n$ since Y_n is a parent of U_m by construction.

We now arrange the conditional expected utilities in a vector as follows.

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

$$\bar{\boldsymbol{U}}_i = (\bar{U}_i(\boldsymbol{y}_{B_i}))^{\mathrm{T}}_{\boldsymbol{y}_{B_i} \in \boldsymbol{\mathcal{Y}}_{B_i}}. \tag{7}$$

2.3. Polynomial structure of expected utility

Generalizing the work in [18] and [21], we introduce a symbolic representation of both the probabilities and the utilities of an MID. For $i \in \mathbb{V}$, $j \in [m]$, $y \in \mathcal{Y}_i$, $\pi \in \boldsymbol{\mathcal{Y}}_{\Pi_i}$ and $\sigma \in \boldsymbol{\mathcal{Y}}_{P_j}$, we define the parameters

$$p_{iy\pi} = P(Y_i = y \mid \boldsymbol{Y}_{\Pi(i)} = \pi), \qquad \psi_{j\sigma} = U_j(\sigma).$$

The first index of $p_{iy\pi}$ and $\psi_{j\sigma}$ refers to the random variable and utility vertex respectively. The second index of $p_{iy\pi}$ relates to the state of the 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 Figure 1 corresponds to $P(Y_6 = 1 \mid Y_5 = 0, Y_4 = 1)$. The *probability* and *utility vectors* are given by $\boldsymbol{p}_i = (p_{iy\pi})^{\mathrm{T}}_{y \in \mathcal{Y}_i, \pi \in \boldsymbol{\mathcal{Y}}_{\Pi_i}}$ and $\boldsymbol{\psi}_j = (\psi_{j\pi})^{\mathrm{T}}_{\pi \in \boldsymbol{\mathcal{Y}}_{P_j}}$, respectively. Parameters are listed within $\boldsymbol{p}_i$ and $\boldsymbol{\psi}_j$ according to a reverse lexicographic order over their indices [see e.g. 38].

Example 5. The symbolic parametrization of the MID in Figure 1 is summarized in Table 1. This is completed by the definition of the criterion weights k_i and h as in equations (1)-(2). In Appendix B 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 equation (2) on the criterion weights. However in the following, unless otherwise indicated, we suppose that all the parameters are unconstrained. Any unmodelled constraint can be added subsequently when investigating the geometric features of the optimal decision.

In the above parametrization, $\bar{\boldsymbol{U}}_i$ consists of a vector of polynomials with 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 between Y_i and U_j (including Y_i) in the DS of G. Then $\bar{\boldsymbol{U}}_i$ 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*

$$r_{iba} = \binom{a-l}{b-l} \prod_{j=i}^{j_a} r_j, \qquad d_{iba} = (b-l) + 2(b-l+1) + w_{ia}, \qquad m_{iba} = h^{b-l} m'_{iba}, \tag{8}$$

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. We say that equation (8) defines the *structure* of the polynomials of the conditional expected utility $\bar{\boldsymbol{U}}_i$. Specifically, the structure of a polynomial consists of its number of monomials and the number of monomials having a certain degree. In Section 5 we show that the entries of the conditional expected utility vectors of several different MIDs share the same polynomial structure. Since additive utility factorizations can be seen as special cases of multiplicative ones by setting $h = 0$, it follows that the conditional expected utility polynomials of an additive ID are square-free.

Corollary 1. *In the notation of Theorem 1, the conditional expected utility $\bar{\boldsymbol{U}}_i$, $i \in [n]$, 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 $r_{ia} = \prod_{j=i}^{j_a} r_j$.*

$\boldsymbol{p}_2 = (p_{211}, p_{201}, p_{210}, p_{200})^{\mathrm{T}}$
$\boldsymbol{p}_3 = (p_{3111}, p_{3011}, p_{3101}, p_{3001}, p_{3110}, p_{3010}, p_{3100}, p_{3000})^{\mathrm{T}}$
$\boldsymbol{p}_5 = (p_{5111}, p_{5011}, p_{5101}, p_{5001}, p_{5110}, p_{5010}, p_{5100}, p_{5000})^{\mathrm{T}}$
$\boldsymbol{p}_6 = (p_{6111}, p_{6011}, p_{6101}, p_{6001}, p_{6110}, p_{6010}, p_{6100}, p_{6000})^{\mathrm{T}}$
$\boldsymbol{\psi}_1 = (\psi_{11}, \psi_{10})^{\mathrm{T}}$, $\boldsymbol{\psi}_2 = (\psi_{21}, \psi_{20})^{\mathrm{T}}$, $\boldsymbol{\psi}_3 = (\psi_{311}, \psi_{301}, \psi_{310}, \psi_{300})^{\mathrm{T}}$

Table 1: Parameterization associated to the MID in Figure 1.

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

Example 6. For the MID of Figure 1 the polynomial structure of the entries of $\bar{\boldsymbol{U}}_5$ can be constructed as follows. From $B_5 = \{3, 4\}$ it follows that $c_5 = 4$. Thus, $\bar{\boldsymbol{U}}_5$ is a column vector of dimension 4. From $U_2 \equiv U_l$ it follows that

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

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 $\bar{\boldsymbol{U}}_5$ 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. The entry $\bar{U}_5(y_3, y_4)$ with $y_3, y_4 = 0, 1$, of this conditional expected utility can be written as $\bar{U}_5(y_3, y_4) = \bar{U}_5^l(y_3, y_4) + \bar{U}_5^{ml}(y_3, y_4)$ where

$$\bar{U}_5^l(y_3, y_4) = k_2(\psi_{21} p_{51 y_4 y_3} + \psi_{20} p_{50 y_4 y_3}) + \sum_{y_5 = 0,1} k_3(\psi_{31 y_4} p_{61 y_5 y_4} + \psi_{30 y_4} p_{60 y_5 y_4}) p_{5 y_5 y_4 y_3}, \tag{9}$$

$$\bar{U}_5^{ml}(y_3, y_4) = h k_2 k_3((\psi_{31 y_4} p_{610 y_4} + \psi_{30 y_4} p_{600 y_4}) \psi_{20} p_{50 y_4 y_3} + (\psi_{31 y_4} p_{611 y_4} + \psi_{30 y_4} p_{601 y_4}) \psi_{21} p_{51 y_4 y_3}), \tag{10}$$

An algorithm for computing the polynomials in Theorem 1 is presented in Section 3.

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 can then be simply 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 their number of monomials can decrease dramatically. We present in Section 3 a plausible numerical specification of the probabilities associated with the MID in Figure 1.

3. The symbolic algorithm

Computing the polynomials in Theorem 1 is a well known NP-hard problem [39]. Here we develop an algorithm based on three operations which exploit the polynomial structure of expected utilities and use only linear algebra calculus. The MapleTH code for their implementation is in Appendix B[6]. Differently from other probabilistic symbolic algorithms [e.g. 13, that computes every possible monomial associated to a model's parametrization and then drops the unnecessary ones], our algorithm sequentially computes only monomials that are part of the MID's expected utilities.

[6] Note that some of the inputs in our MapleTH functions are different from the ones reported here. Such inputs are chosen to illustrate the procedure as concisely as possible.

Algorithm 3.1: 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**
  - $s_k = \prod_{l=k+1}^{j} \mathbb{1}_{\{l \in \{B_{i+1} \cup P_j\}\}}(r_l)$
  - **if** $k \in B_{i+1}$
  - **then** $\left\{ \boldsymbol{\psi}_j = \left(\underbrace{\boldsymbol{\psi}_j^{s_k,1} \quad \cdots \quad \boldsymbol{\psi}_j^{s_k,1}}_{r_k \text{ times}} \quad \cdots \quad \underbrace{\boldsymbol{\psi}_j^{s_k,c_j/s_k} \quad \cdots \quad \boldsymbol{\psi}_j^{s_k,c_j/s_k}}_{r_k \text{ times}} \right) \right.$
  - **else if** $k \in P_j$
  - **then** $\left\{ \bar{\boldsymbol{U}}_{i+1} = \left(\underbrace{\bar{\boldsymbol{U}}_{i+1}^{s_k,1} \quad \cdots \quad \bar{\boldsymbol{U}}_{i+1}^{s_k,1}}_{r_k \text{ times}} \quad \cdots \quad \underbrace{\bar{\boldsymbol{U}}_{i+1}^{s_k,c_i/s_k} \quad \cdots \quad \bar{\boldsymbol{U}}_{i+1}^{s_k,c_i/s_k}}_{r_k \text{ times}} \right) \right.$

return $(\bar{\boldsymbol{U}}_{i+1}, \boldsymbol{\psi}_j)$

3.1. A new algebra for MIDs

We need to introduce two simple procedures, `EUDuplicationPsi` and `EUDuplicationP`, entailing a change of dimension of probability, utility and conditional expected utility vectors. These are required in order to multiply parameters associated to compatible instantiations only. For conciseness, we detail here only `EUDuplicationPsi` and refer to Appendix B for the code of both procedures.

The steps of `EUDuplicationPsi` are shown in Algorithm 3.1. For a vector $\boldsymbol{\psi}$, let $\boldsymbol{\psi}^{s,t}$ be the subvector of $\boldsymbol{\psi}$ 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: a conditional expected utility $\bar{\boldsymbol{U}}_{i+1}$; the utility vector associated to the utility node preceding Y_{i+1}, $\boldsymbol{\psi}_j$; 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 $\boldsymbol{r} = (r_1, \ldots, r_n)^{\mathrm{T}}$.

For all indices smaller than i and not in $B_{i+1} \cap P_j$, Algorithm 3.1 computes a positive integer number s_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 s_k rows of $\boldsymbol{\psi}_j$ 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 a conditional expected utility. In the algorithm of Section 3.3, an `EUMultiSum` operation is associated to every utility vertex of the MID. Let $P = \{P_1, \ldots, P_m\}$.

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

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

The second operation, `EUMarginalization` is applied to any random vertex of the MID.

Definition 5 (EUMarginalization). For $i \in \mathbb{V}$, let $\bar{\boldsymbol{U}}_{i+1}$ be a conditional expected utility vector and $\boldsymbol{p}_i$ a probability vector. The EUMarginalization, Σ^{EU}, between $\bar{\boldsymbol{U}}_{i+1}$ and $\boldsymbol{p}_i$ is defined as

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

$k_1 = 0.2,$	$h = 2.6,$	$\psi_{311} = 0,$	$p_{5101} = 0.9$	
$k_2 = 0.3,$	$\psi_{20} = 1,$	$\psi_{300} = 1,$	$p_{5110} = 0.2,$	$p_{6100} = 0.3$
$k_3 = 0.5,$	$\psi_{21} = 0,$	$\psi_{310} = 0.4,$	$p_{5100} = 0.6$	

Table 2: Numerical specification of a subset of the unknown variables associated to the MID of Figure 1.

2. $I_i' \times (\bar{\boldsymbol{U}}_{i+1}' \circ \boldsymbol{p}_i')$, where $\times$ is the standard matrix product and I_i' is a matrix with $c_{i+1}s_i/r_i \in \mathbb{Z}_{\geq 1}$[7] rows and $c_{i+1}s_i$ columns defined as

$$I_i' = \left(\begin{pmatrix} \mathbf{1} & \mathbf{0} & \cdots & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{0} & \mathbf{1} & \cdots & \mathbf{0} \end{pmatrix} \cdots \begin{pmatrix} \mathbf{0} & \mathbf{0} & \cdots & \mathbf{1} \end{pmatrix} \right)^{\mathrm{T}}$$

where $\mathbf{1}$ and $\mathbf{0}$ 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 maximization of $\bar{\boldsymbol{U}}_{i+1}$ over the space $\mathcal{Y}_i$, $i \in \mathbb{D}$, for any element of $\boldsymbol{\mathcal{Y}}_{\Pi(i)}$.

Definition 6 (EUMaximization). For $i \in \mathbb{D}$, let $\bar{\boldsymbol{U}}_{i+1}$ be a conditional expected utility vector. An EUMaximization over $\mathcal{Y}_i$, $\max_{\mathcal{Y}_i}^{EU}$, is defined by the following steps:

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

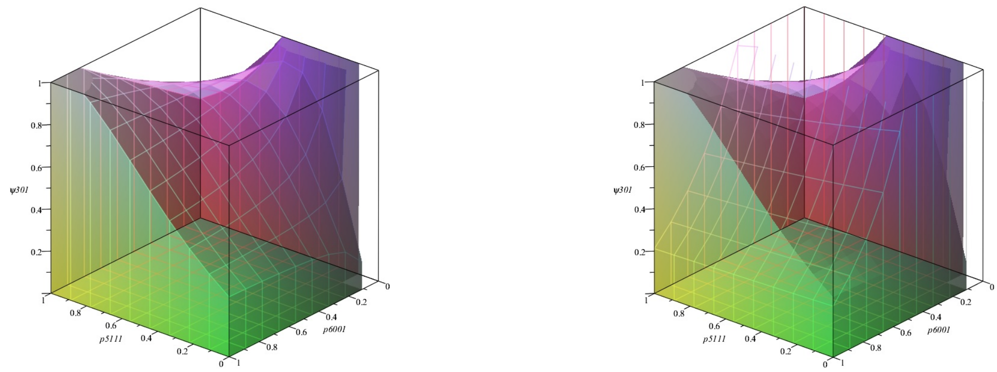
$$I_i^* = \left(\begin{pmatrix} \boldsymbol{e}_{\boldsymbol{y}_i^*(1)} & \mathbf{0} & \cdots & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{0} & \boldsymbol{e}_{\boldsymbol{y}_i^*(2)} & \cdots & \mathbf{0} \end{pmatrix} \cdots \begin{pmatrix} \mathbf{0} & \mathbf{0} & \cdots & \boldsymbol{e}_{\boldsymbol{y}_i^*(c_{i+1}/r_i)} \end{pmatrix} \right)^{\mathrm{T}}$$

where $\boldsymbol{e}_{\boldsymbol{y}_i^*(\pi)}$, $\pi \in [c_{i+1}/r_i]$, 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.

The first item in Definition 6 is critical for `EUMaximization`. It is not the scope of this paper to present a methodology to identify such expected utility maximizing decisions. We simply assume that these can be found. However we note that, within our symbolic approach, polynomial optimization and semi-algebraic methods can be used to determine such optimal decisions [40]. Once these are identified, `EUMaximization` drops the polynomials associated with the non optimal course of action. Alternatively, we can think of the evaluation of the MID as the computation of the expected utility polynomial of a specific policy. This is for example all that can be done whenever decision rules have been committed to a priori. In this case, `EUMaximization` deletes the decisions that are not adopted within the given policy. The MapleTH function `EUMaximization` in Appendix B calls a subfunction `Maximize`, which randomly picks optimal decisions.

Example 7. To illustrate the working of `EUMaximization` assume for the MID in Figure 1 that a DM has provided the specifications summarized in Table 2 together with the *qualitative* beliefs $p_{5111} = p_{6011}$ and $p_{6010} = p_{6001}$. These in general cannot be implemented in a non-symbolic approach to decision making problems. By plugging in these numerical values and constraints into equations (9) and (10), the DM would choose to evacuate for combination of parameters denoted by the coloured regions of Figure 2. The geometric structure of these regions often gives insights about the maximization process. Assuming $Y_3 = 1$, if the DM believes that $\psi_{301}, p_{5111}, p_{6001} \in [0, 1/2]$, then Figure 2a shows that evacuation will be the optimal choice. Conversely, when $Y_3 = 0$, such a range for the indeterminates would not uniquely identify an optimal course of action. We can however envisage the algorithm to work over the two sub-regions of Figure 2b separately. The algorithm would then output different optimal courses of action for different combinations of the unknown parameters. We plan to develop a systematic methodology to address these issues in later work.

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



(a) Optimal regions in the case $Y_3 = 1$. (b) Optimal regions in the case $Y_3 = 0$.

Figure 2: Regions determining the combinations of parameters leading to an optimal decision of evacuating (coloured regions) and not evacuating (white regions) for the evaluation of the diagram in Figure 1 given the partial numeric specification in Table 2.

3.2. Polynomial interpretation of the operations

Each of the above three operations changes the conditional expected utility vectors and their entries in a specific way formalized in Proposition 2.

Proposition 2. *For $i \in [n-1]$, let $\bar{\boldsymbol{U}}_{i+1}$ be a conditional expected utility vector whose entries have the polynomial structure of equation (8) and let U_j be the vertex preceding Y_{i+1} in the DS. Then in the notation of Theorem 1*

- $\max^{EU}_{\mathcal{Y}_i} \bar{\boldsymbol{U}}_{i+1}$ *has dimension $c_{i+1}/r_i \in \mathbb{Z}_{\geq 1}$ and its entries do not change polynomial structure;*
- $\bar{\boldsymbol{U}}_{i+1} +^{EU} \boldsymbol{\psi}_j$ *has dimension $c_{i+1}s_i^U$, where $s_i^U = \prod_{k\in\{P_j \setminus B_{i+1}\}} r_k$, 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;*
- $\bar{\boldsymbol{U}}_{i+1}\Sigma^{EU}\boldsymbol{p}_i$ *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 whose effect on the polynomials associated to the diagram in Figure 1 is illustrated below.

3.3. A fast algorithm for MIDs' evaluations

The algorithm for the evaluation of MIDs is given in Algorithm 3.2. This receives as input the DS of the MID, S say, the sets $\mathbb{J}$, $\mathbb{V}$ and $\mathbb{D}$, and the vectors $\boldsymbol{p} = (\boldsymbol{p}_1, \ldots, \boldsymbol{p}_n)^{\mathrm{T}}$, $\boldsymbol{\psi} = (\boldsymbol{\psi}_1, \ldots, \boldsymbol{\psi}_m)^{\mathrm{T}}$ and $\boldsymbol{k} = (k_1, \ldots, k_m, h)^{\mathrm{T}}$. This corresponds to a symbolic version of the backward induction procedure 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.

In line (1) the conditional expected utility $\bar{\boldsymbol{U}}_{n+1}$ is initialized to (0). Lines (2) and (3) index a reverse loop over the indices of both 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 $\boldsymbol{p}_k$, jointly to an `EUMultiSum` with $\boldsymbol{\psi}_l$. In the other case, `EUMaximization` and `EUMarginalization` operations are performed without `EUMultiSum`. The MapleTH function `SymbolicExpectedUtility` in Appendix B is an implementation of Algorithm 3.2.

Algorithm 3.2: SYMBOLICEXPECTEDUTILITY$(\mathbb{J}, S, \boldsymbol{p}, \boldsymbol{\psi}, \boldsymbol{k}, \mathbb{V}, \mathbb{D})$

$\bar{\boldsymbol{U}}_{n+1} = (0)$
for $k \leftarrow n$ **downto** 1
do {
  for $l \leftarrow m$ **downto** 1
  do {
    if $k = j_l$
    then {
      if $k \in \mathbb{D}$
      then $\{\bar{\boldsymbol{U}}_k = \max^{EU}_{\mathcal{Y}_k}(\bar{\boldsymbol{U}}_{k+1} +^{EU} \boldsymbol{\psi}_l)$
      else $\{\bar{\boldsymbol{U}}_k = \boldsymbol{p}_k\ \Sigma^{EU}\ (\bar{\boldsymbol{U}}_{k+1} +^{EU} \boldsymbol{\psi}_l)$
    else if $k \in \mathbb{D}$
    then $\{\bar{\boldsymbol{U}}_k = \max^{EU}_{\mathcal{Y}_k} \bar{\boldsymbol{U}}_{k+1}$
    else $\{\bar{\boldsymbol{U}}_k = \boldsymbol{p}_k\ \Sigma^{EU}_{\mathcal{Y}_k}\ \bar{\boldsymbol{U}}_{k+1}$
return $(\bar{\boldsymbol{U}}_1)$

Example 8. For the MID in Figure 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

$$P_3 = \{4, 6\}, \quad s^U_6 = 4, \quad \Pi_6 = \{4, 5\}, \quad s_6 = 2.$$

Thus, first $\bar{\boldsymbol{U}}_7$ is replicated four times (since $s^U_6 = 4$) through the function `EUDuplicationPsi` and

$$\bar{\boldsymbol{U}}_7 +^{EU} \boldsymbol{\psi}_3 = \begin{pmatrix} k_3\psi_{11} & k_3\psi_{01} & k_3\psi_{10} & k_3\psi_{00} \end{pmatrix}^{\mathrm{T}}. \tag{11}$$

Then, the right hand side (rhs) of equation (11) is duplicated via `EUDuplicationP` (since $s_6 = 2$) and

$$\bar{\boldsymbol{U}}_6 = I'_6 \times \bar{\boldsymbol{U}}^t_6 \circ \boldsymbol{p}_6 = (k_3\psi_{31j}p_{61ij} + k_3\psi_{30j}p_{60ij})^{\mathrm{T}}_{i,j=0,1}, \tag{12}$$

where $\bar{\boldsymbol{U}}^t_6$ is equal to the duplicated version of the right hand side of equation (11). The vector $\bar{\boldsymbol{U}}_6$ 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 first called which gives as output

$$\bar{\boldsymbol{U}}^t_5 \triangleq \bar{\boldsymbol{U}}_6 +^{EU} \boldsymbol{\psi}_2 = h \cdot k_2 \cdot \bar{\boldsymbol{U}}_6 \circ \begin{pmatrix} \psi_{21} & \psi_{20} & \psi_{21} & \psi_{20} \end{pmatrix}^{\mathrm{T}} + \bar{\boldsymbol{U}}_6 + k_2 \cdot \begin{pmatrix} \psi_{21} & \psi_{20} & \psi_{21} & \psi_{20} \end{pmatrix}^{\mathrm{T}} \tag{13}$$

The polynomial $\bar{\boldsymbol{U}}^t_6$ is the sum of two monomials of degree 3 inherited from $\bar{\boldsymbol{U}}_6$, of two monomials of degree 6 (from the first term on the rhs of equation (13)) and one monomial of degree 2 (from the last term on the rhs of equation (13)). Its dimension is equal to four since $c_6 = 4$ and $s^U(5) = 0$ (that is, no `EUDuplicationPsi` is required). Thus, `EUMultiSum` manipulates the conditional expected utility vector according to Proposition 2. Then the `EUMarginalization` function computes $\bar{\boldsymbol{U}}_5 = I'_5 \times \left(\begin{pmatrix} \bar{\boldsymbol{U}}^t_5 & \bar{\boldsymbol{U}}^t_5 \end{pmatrix}^{\mathrm{T}} \circ \boldsymbol{p}_5\right)$. Each entry of $\bar{\boldsymbol{U}}_5$ has twice the number of monomials of the entries of $\bar{\boldsymbol{U}}^t_5$ and each monomial of $\bar{\boldsymbol{U}}_5$ has degree $d+1$, where d is the degree of each monomial of $\bar{\boldsymbol{U}}^t_5$ (whose entries are homogeneous polynomials). These vectors also have the same dimension since $s_5 = 2$ and $r_5 = 2$. Thus, this `EUMarginalization` changes the conditional expected utility vector according to Proposition 2. The polynomial in a generic entry of $\bar{\boldsymbol{U}}_5$ was shown in equations (9)-(10).

The algorithm then considers the controlled variable Y_4. Since $4 \notin \mathbb{J}$, Y_4 is not the argument of a utility function with the highest index and therefore the algorithm calls the `EUMaximization` function. Suppose the optimal decisions are identified to be $Y_4 = 1$ when $Y_3 = 1$ and $Y_4 = 0$ when $Y_3 = 0$. The evaluation would then suggest that the population is evacuated whenever a high level of deposition is observed and that people are not evacuated if the deposition is low. Thus, the function returns $\bar{\boldsymbol{U}}_4 = I_4^* \times \bar{\boldsymbol{U}}_5$, where I_4^* is a 2×4 matrix with ones in positions $(1,1)$ and $(2,4)$ and zeros otherwise. Proposition 2 is respected since the entries of $\bar{\boldsymbol{U}}_4$ have the same polynomial structure of those of $\bar{\boldsymbol{U}}_5$ and $\bar{\boldsymbol{U}}_4$ has dimension 2.

The `SymbolicExpectedUtility` function then applies in sequence the operations defined in Section 3.1. For the MID in Figure 1 this sequentially computes the following quantities:

$$\bar{\boldsymbol{U}}_3^t = h \cdot k_1 \cdot \bar{\boldsymbol{U}}_4 \circ \boldsymbol{\psi}_1 + \bar{\boldsymbol{U}}_4 + k_1 \cdot \boldsymbol{\psi}_1, \qquad \bar{\boldsymbol{U}}_3 = I_3^{'} \times \left(\begin{pmatrix} \bar{\boldsymbol{U}}_3^t & \bar{\boldsymbol{U}}_3^t & \bar{\boldsymbol{U}}_3^t & \bar{\boldsymbol{U}}_3^t \end{pmatrix}^{\mathrm{T}} \circ \boldsymbol{p}_3 \right),$$
$$\bar{\boldsymbol{U}}_2 = I_2^{'} \times \left(\bar{\boldsymbol{U}}_3 \circ \boldsymbol{p}_2 \right), \qquad \bar{\boldsymbol{U}}_1 = \begin{pmatrix} 1 & 0 \end{pmatrix} \times \bar{\boldsymbol{U}}_2,$$

assuming the optimal initial decision is $Y_1 = 1$.

Interestingly, using the new algebra we introduced in Section 3.1, the evaluation of an MID can be written as a simple algebraic expression. For example, the evaluation of the MID in Figure 1 can be written as

$$\bar{\boldsymbol{U}}_1 = \max\nolimits_{\mathcal{Y}_1}^{EU} \left(\boldsymbol{p}_2 \, \Sigma^{EU} \left(\boldsymbol{p}_3 \, \Sigma^{EU} \left(\boldsymbol{\psi}_1 +^{EU} \max\nolimits_{\mathcal{Y}_4}^{EU} \left(\boldsymbol{p}_5 \, \Sigma^{EU} (\boldsymbol{\psi}_2 +^{EU} (\boldsymbol{p}_6 \, \Sigma^{EU} \boldsymbol{\psi}_3)) \right) \right) \right) \right)$$

and this polynomial can be evaluated with `SymbolicExpectedUtility`.

4. Modifying the topology of the MID

Algorithm 3.2 works under the assumption that the MID is in extensive form. In practice 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 4.1 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 4.2 we describe the polynomial interpretation of a criterion introduced in [37] and [41] to identify a subgraph of the original MID whose associated optimal decision rule is the same as the one of the original MID.

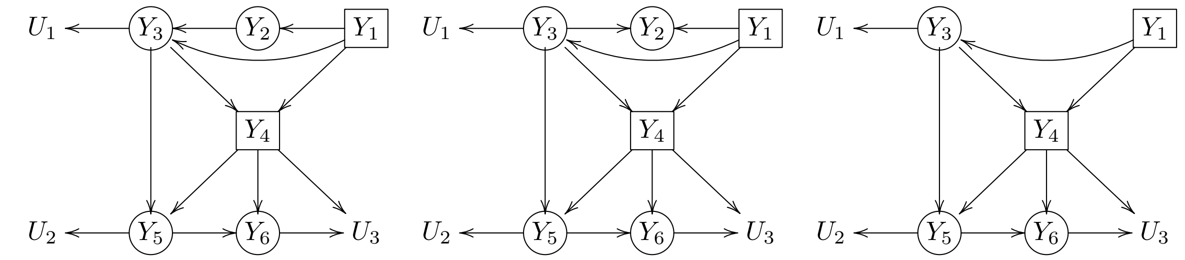
4.1. Rules to transform an MID in extensive form

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 [7].

Example 9. The MID in Figure 3a is a non-extensive variant of the MID in Figure 1 not including the edge (Y_2, Y_4). The MID in Figure 3b is obtained by the reversal of the edge (Y_2, Y_3) and the MID in Figure 3c is the network in extensive form obtained by deleting the barren node Y_2.

Let Y_i be a father of Y_j and Y_j a son of Y_i 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.

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:*



(a) A variant of the MID of Figure 1 without (Y_2, Y_4)

(b) MID resulting from the reversal of the arc (Y_2, Y_3) from the MID of Figure 3a.

(c) MID resulting from the removal of the barren node Y_2 from the MID of Figure 3b and from the application of the sufficiency theorem to the MID of Figure 1

Figure 3: Example of a sequence of manipulations of a non extensive form MID.

- ***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*

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

- ***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:*

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

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 3.2, and similarly the `SymbolicExpectedUtility` MapleTH function, 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 3.2 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 3.2 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 $\bar{\boldsymbol{U}}'_j$ is the conditional expected utility associated to G' after the marginalization of Y_i and $\bar{\boldsymbol{U}}'_i$ is the conditional expected utility 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 polynomial 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 $\pi_i \in \boldsymbol{\mathcal{Y}}_{\Pi_i}$, $\pi_j \in \boldsymbol{\mathcal{Y}}_{\Pi_j}$, $\pi'_i \in \boldsymbol{\mathcal{Y}}_{\Pi'_i}$, $\pi'_j \in \boldsymbol{\mathcal{Y}}_{\Pi'_j}$, $y_i \in \mathcal{Y}_i$ 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 $\Pi^i_j = \Pi_j \setminus \{i\}$. Then*

– *in the new parametrization* $\boldsymbol{p}'_i$ *is deleted;*
– *in the original parametrization* $\boldsymbol{p}_i$ *is deleted and* $p_{jy_j\pi^i_j0} = \cdots = p_{jy_j\pi^i_jr_i-1}$, *for* $y_j \in \mathcal{Y}_j$, $\pi^i_j \in \boldsymbol{\mathcal{Y}}_{\Pi^i_j}$, *where the fourth index of* $p_{jy_j\pi^i_ji}$, $i \in [r_i-1]$, *refers to the instantiation of* Y_i.

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

Example 10. Reversing the edge (Y_2, Y_3) in the MID of Figure 3a, 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 in Figure 3b simply corresponds to canceling the vectors $\boldsymbol{p}_2$ and $\boldsymbol{p}'_2$ and setting $p_{3y_31y_1}$ equal to $p_{3y_30y_1}$ for any $y_1, y_3 \in \{0, 1\}$.

A consequence of Proposition 4 is that manipulations of the diagram change the polynomial structure of the conditional expected utilities under the new parametrization $\boldsymbol{p}'$. 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 of the MID.

Lemma 1. *Under the assumptions of Proposition 4 and in the notation of Theorem 1 the following holds:*

1. *let* x *be the smallest index in* $\Pi_i \cup \Pi_j$, *reverse of the arc* (Y_i, Y_j) *and evaluate the MID using the adjusted Algorithm 3.2:*
   (a) *if* $j \notin \mathbb{J}$, *then*
      i. *the entries of* $\bar{\boldsymbol{U}}'_j$ *have* $r_ir_{jba}/r_j \in \mathbb{Z}_{\geq 1}$[8] *monomials of degree* d_{jba}*; for* $i < k < j$, *the entries of* $\bar{\boldsymbol{U}}'_k$ *can have different polynomial structure from the ones of* $\bar{\boldsymbol{U}}_k$ *according to Proposition 2;*
      ii. *the vectors* $\bar{\boldsymbol{U}}'_k$, $x < k \leq j$, *have dimension* $c'_k = \prod_{s\in\{B'_k\setminus\{k,\dots,n\}\}} r_s$ *where* $B'_k = B_k \cup \{l : (Y_l, Y_i)$ *or* $(Y_l, Y_j) \in E(G')\}$*;*
   (b) *if* $j \in \mathbb{J} \cap \mathbb{J}'$, *then*
      i. *the entries of* $\bar{\boldsymbol{U}}'_j$ *have* $r_ir_{(j+1)ba}$ *monomials of degree* $d_{(j+1)ba} + 1$ *and, for* $i < k < j$, *the entries of* $\bar{\boldsymbol{U}}'_k$ *have a different polynomial structure from the ones of* $\bar{\boldsymbol{U}}_k$ *according to Proposition 2;*
      ii. *for* $x < k < j$, $\bar{\boldsymbol{U}}'_k$ *has dimension* $c''_k = \prod_{s\in\{B''_k\setminus\{k,\dots,n\}\}} r_s$, *with* $B''_k = B'_k \cup P_{j_j}$*;*
   (c) *if* $j \notin \mathbb{J}'$, *suppose* $j \in P_t$ *and* s *is the second highest index in* P_t, *then*
      i. *for* $s < k \leq j$, *the entries of* $\bar{\boldsymbol{U}}'_k$ *have the polynomial structure specified in point 1.(b) and dimension* c''_k*;*
      ii. $i < k \leq s$, *the entries of* $\bar{\boldsymbol{U}}'_k$ *have the polynomial structure specified in point 1.(a) and dimension* c'_k.
      iii. *for* $x < k \leq i$, $\bar{\boldsymbol{U}}'_k$ *has dimension* c'_k *and the polynomial structure of its entries does not change;*
2. *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*
   (a) *for* $i < k \leq z$, $\bar{\boldsymbol{U}}'_k$ *has* c'_k/r_i *entries whose polynomial structure does not change;*
   (b) *for* $k \leq i$, $\bar{\boldsymbol{U}}'_k$ *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 11. After the reversal of the edge (Y_2, Y_3) from the network in Figure 3a, the polynomial structure of the conditional expected utilities associated to the original and to the manipulated diagrams is reported in Table 3 by $\bar{\boldsymbol{U}}_i$ and $\bar{\boldsymbol{U}}^r_i$ respectively. Since Y_3 is the only argument of U_1 we are in Item (1b) of Lemma 1. The conditional expected utility $\bar{\boldsymbol{U}}^r_3$ is obtained running the adjusted Algorithm 3.2 for the network of Figure 3b

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

$\bar{\boldsymbol{U}}_2 \equiv \bar{\boldsymbol{U}}_1$			$\bar{\boldsymbol{U}}_3$			$\bar{\boldsymbol{U}}_4$			$\bar{\boldsymbol{U}}_3^r$			$\bar{\boldsymbol{U}}_2^r \equiv \bar{\boldsymbol{U}}_1^r$			$\bar{\boldsymbol{U}}_3^b \equiv \bar{\boldsymbol{U}}_1^b$		
#	d.	s.f.	#	d.	s.f.	#	d.	s.f.	#	d.	s.f.	#	d.	s.f.	#	d.	s.f.
4	4	yes	2	3	yes	2	3	yes	4	4	yes	4	4	yes	2	3	yes
8	5	yes	4	4	yes	4	4	yes	8	5	yes	8	5	yes	4	4	yes
16	6	yes	8	5	yes	4	7	yes	8	8	yes	16	6	yes	8	5	yes
8	8	yes	4	7	yes							8	8	yes	4	7	yes
32	9	yes	16	8	yes							32	9	yes	16	8	yes
16	12	no	8	11	no							16	12	no	8	11	no

Table 3: Polynomial structure of the conditional expected utilities for the original MID, $\bar{\boldsymbol{U}}_j$, for the one after the reversal of the arc (Y_2, Y_3), $\bar{\boldsymbol{U}}_j^r$ and for the one after the removal of the barren node Y_2, $\bar{\boldsymbol{U}}_j^b$. The symbol # corresponds to the number of monomials, d. to the degree and s.f. whether or not they are square free.

after the marginalization of Y_2. This can be noted to change according to Lemma 1, by comparing its structure to the one of $\bar{\boldsymbol{U}}_4$. Furthermore, $\bar{\boldsymbol{U}}_2^r$ and $\bar{\boldsymbol{U}}_1^r$ have the same polynomial structures as $\bar{\boldsymbol{U}}_2$ and $\bar{\boldsymbol{U}}_1$. The last 3 columns of the Table 3 show the polynomial structure of the conditional expected utilities $\bar{\boldsymbol{U}}_3^b$ associated to the MID in Figure 3c which does not include Y_2. According to Lemma 1, $\bar{\boldsymbol{U}}_3^b$ has the same polynomial structure of $\bar{\boldsymbol{U}}_3$ and for each row of the table, the number of monomials with degree d in $\bar{\boldsymbol{U}}_1^b$ is half the number of monomials of $\bar{\boldsymbol{U}}_1$ having degree $d+1$.

4.2. The sufficiency principle

After an MID has been transformed in extensive form according to the rules in Section 4.1, 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 DAG [42]. In order to state the d-separation criterion, we need to introduce a few concepts of 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. For any three disjoint subvectors $\boldsymbol{Y}_A, \boldsymbol{Y}_B, \boldsymbol{Y}_C \in V(G^M)$, $\boldsymbol{Y}_A$ is d-separated from $\boldsymbol{Y}_C$ by $\boldsymbol{Y}_B$ in G^M if and only if any path from any vertex $Y_a \in \boldsymbol{Y}_A$ to any vertex $Y_c \in \boldsymbol{Y}_C$ passes through a vertex $Y_b \in \boldsymbol{Y}_B$ in $\mathcal{S}(G^M)$.

Proposition 5. *Let $j \in \mathbb{D}$ and $i \in \mathbb{V} \cap \Pi_j$. 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, then 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, letting Ch_i be the index set of the children of Y_i,*

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

The sufficiency principle can be stated for a vector of variables [e.g. 37, 41]. However, we can simply apply the criterion in Proposition 5 for each variable of the vector and obtain the same result.

Example 12. The MID in Figure 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 in Figure 3c. Exceptionally, this is 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 the sufficiency principle changes the symbolic parametrization of the MID.

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 a father of Y_k and a parent (not a father) of Y_j in G. Let Π'_k be the parent set of a vertex Y_k in G' and for $l \in [n]$*

$$\Pi_k^{>i} = \Pi(k) \setminus [i-1], \qquad \Pi_i^{k,l} = \Pi_l \cap \Pi_k \cap \Pi_i.$$

Then the reparametrization of the MID with graph G' is

$$p'_{ky_k\pi'_k} = \sum_{y_i \in \mathcal{Y}_i} p_{ky_k\pi_k} p_{iy_i\pi_i}, \qquad p'_{jy_j\pi'_j} = \sum_{y_j \in \mathcal{Y}_j} p_{jy_j\pi_j} \frac{\prod_{l \in \Pi_j^{>i}} \sum_{y_{\Pi_i^{j,l}} \in \boldsymbol{\mathcal{Y}}_{\Pi_i^{j,l}}} p_{ly_l\pi_l} p_{iy_i\pi_i}}{\sum_{y_i \in \mathcal{Y}_i} \prod_{l \in \Pi_j^{>i}} \sum_{y_{\Pi_i^{j,l}} \in \boldsymbol{\mathcal{Y}}_{\Pi_i^{j,l}}} p_{ly_l\pi_l} p_{iy_i\pi_i}}$$

The proof of this proposition is provided in Appendix A.5. Again, this new parametrization $\boldsymbol{p}'$ implies a change in the conditional expected utility vectors.

Lemma 2. *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 conditional expected utility vectors in G' are such that*

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

Proof. Item 1 of Corollary 2 is a straightforward consequence of Proposition 2, since the deletion of the vertex Y_i entails one less EUMarginalization during Algorithm 3.2. Item 2 of Corollary 2 follows from the fact that the sets B_i and B'_i only affect the dimension of the conditional expected utility vectors.

Since the application of the sufficiency principle to the diagram of Figure 3a 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 3.

5. The lattice of equivalent MIDs

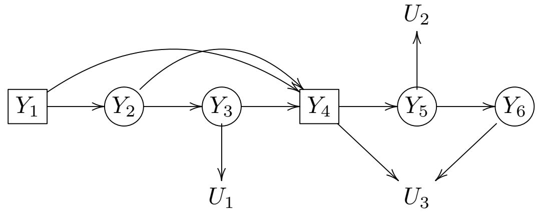
Theorem 1 specifies the polynomial structure of the expected utility associated to any MID at any stage of its evaluation. However, as we show in this section, there are many MIDs whose conditional expected utilities share the same polynomial structure. Such MIDs are called **equivalent**.

5.1. Equivalence between MIDs

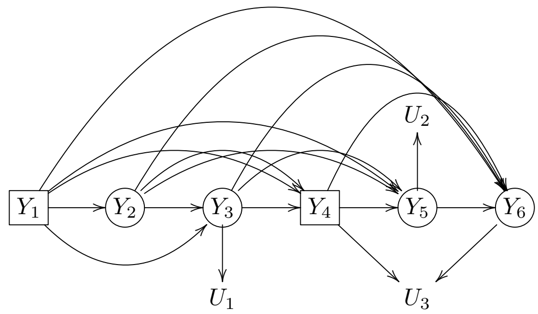
Definition 7. Two MIDs in extensive form are equivalent if, under our parametrization, the following conditions hold:

1. they have the same vertices with the same labelling;
2. they imply the same total order over $\{Y_i : i \in \mathbb{D}\}$;
3. they have the same sets P_j, $j \in [m]$;
4. they imply the same partial order over $\{Y_1, \dots, Y_n\}$

Points 1-3 of Definition 7 specify that the decision problems associated to equivalent MIDs can be modelled by MIDs with the same vertex sets, more specifically that there is a one-to-one mapping between random variables and decision variables (point 1), that the order in which decisions are committed to is equal (point 2) and that the utility functions have the same arguments (point 3). Point 4 implies a simplification which does not limit generality. This is imposed because different partial orders would require a



(a) Minimal MID equivalent to one in Figure 1.



(b) Maximal MID equivalent to the on Figure 1.

Figure 4: Two MIDs in the same equivalence class of the diagram in Figure 1

completely different parameter set. Note however that different partial orders describing the same conditional independence structure are associated to the same probability density factorization and therefore lead to evaluations with the same output [see e.g. 37] In particular the edge sets of equivalent MIDs can differ only for edges into a non controlled variable. Specifically, equivalent MIDs differ at most for the probabilistic structure they imply. This is formalized by Proposition 7.

Proposition 7. *An MID G is equivalent to an MID G' if and only if*

- $V(G) = V(G')$;
- *if* $(Y_i, U_j) \in E(G)$, *for* $j \in [m]$ *and* $i \in [n]$, *then* $(Y_i, U_j) \in E(G')$;
- *if* $(Y_i, Y_j) \in E(G)$, *for* $i \in \mathbb{V}$ *and* $j \in \mathbb{D}$, *then* $(Y_i, Y_j) \in E(G')$.

Example 13. The MIDs in Figures 1 and 4 are equivalent according to Proposition 7.

The equivalence between MIDs of Definition 7 is a proper equivalence relation and conditional expected utilities of MIDs in the same equivalence class $\mathcal{C}$ share the same polynomial structure. This is stated in Proposition 8 whose proof is immediate.

Proposition 8. *All MIDs in the equivalence class $\mathcal{C}$ have conditional expected utility vectors with the same polynomial structure.*

Corollary 2. *The only difference between MIDs in the equivalence class $\mathcal{C}$ might consist of a different dimension for some of their conditional expected utility vectors.*

Proof. This result follows directly from the characterization of equivalence in Proposition 7, since the sets B_i, $i \in [n]$, for two equivalent MIDs are the only objects that can differ and these can only affect the dimension of the conditional expected utility vectors.

The different dimensions of conditional expected utility vectors of equivalent MIDs are actually constrained by a rather strong order over the elements of $\mathcal{C}$. Specifically, this is a lattice as shown in Section 5.3.

5.2. An introduction to lattices

Let $(L, \prec)$ be a partially ordered set, where L denotes a set and $\prec$ is a reflexive, antisymmetric and transitive binary relation. For an $S \subseteq L$, an element $u \in L$ is said to be an *upper bound* of S if $s \prec u$ for every $s \in S$. An upper bound u of S is said to be its least upper bound, or *join*, if $u \prec x$ for each upper bound x of S. Similarly, an $l \in L$ is said to be a *lower bound* of S, if $l \prec s$ for every $s \in S$. A lower bound l of S is said to be a greatest lower bound, or *meet*, if $x \prec l$ for each lower bound x of S.

A *lattice* is a partially ordered set in which every two elements $x, y \in L$ have a join and a meet, denoted by $x \vee y$ and $x \wedge y$ respectively.

Example 14. Let L be a set and $\mathcal{P}(L)$ its power set. The operation of set inclusion gives the structure of a lattice to $\mathcal{P}(L)$ for which the meet of any two sets in $\mathcal{P}(L)$ is their intersection and the join their union.

A lattice is said to be *bounded* if it has an element 1 called maximum and an element 0 called minimum such that $0 \prec x \prec 1$ for all $x \in L$. Note that $x \vee 1 = 1$ and $x \wedge 0 = 0$ for any $x \in L$.

Example 15 (Example 14 continued). The minimum is the empty set and the maximum is the complete set, L.

A lattice is said to be *distributive* if for any $x, y, z \in L$ one of the following two equivalences hold

$$x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z), \qquad x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z), \tag{14}$$

i.e. if one operation distributes over the other.

Example 16 (Example 14 continued). The lattice over the elements of the power set is distributive since both the union and intersection operations respect equation (14).

An element y of the lattice is said to *cover* another element x if $x \prec y$ and there is no other element z of the lattice such that $x \prec z \prec y$. A lattice is called *ranked* if there is a rank function $r : L \to \mathbb{N}$ such that if y covers x then $r(y) = r(x) + 1$. The value of the rank function of an element $x \in L$ is called the rank of x.

Example 17 (Example 14 continued). For $x \in \mathcal{P}(L)$ and any $a \in L \setminus x$, the set $x \cup \{a\}$ covers x. The power set of L is a ranked lattice and a rank function is given by the number of elements of $x \in L$.

The elements $x_1, \dots, x_n$ of L form a *chain* if $x_1 \prec \cdots \prec x_n$. The *length* of the chain is equal to the number of its elements (in this case n). A chain is *maximal* if x_i covers x_{i-1}, for $i = 2, \dots, n$. If for any $x, y \in L$, such that $x \prec y$, any maximal chain from x to y has the same length, then the lattice is said to respect the *Jordan-Dedeking chain condition.*

Example 18 (Example 14 continued). The power set respects the Jordan-Dedeking chain condition since for any two subsets $x, y \in \mathcal{P}(L)$, $x \prec y$, a maximal chain from x to y consists to a sequential union of an element in $y \setminus x$, where the order in which elements are added is irrelevant.

5.3. Lattice structure within an equivalence class

In this section the lattice structure of an equivalence class $\mathcal{C}$ of MIDs is studied.

Definition 8. We say that

1. An MID $G_{min} \in \mathcal{C}$ is *minimal* when, for any $i \in [n]$ and $j \in \mathbb{V}$, $(Y_i, Y_j) \in E(G_{min})$ if and only if Y_j covers Y_i in the partial order over $\{Y_1, \dots, Y_n\}$. The number of edges in $E(G_{min})$ is n_{min}.
2. An MID $G_{max} \in \mathcal{C}$ is *maximal* if, for any $i \in [n]$, $(Y_j, Y_i) \in E(G_{max})$ for all $j \in [i-1]$. The number of edges in $E(G_{max})$ terminating into random vertices is n_{max}.

The minimal MID can be thought of as the MID including the highest number of conditional independence statements respecting the partial order over $\{Y_1, \dots, Y_n\}$. The maximal MID on the other hand can be seen as a saturated model, implying no conditional independence statements. The minimal and maximal MIDs are particularly important since all the other elements of $\mathcal{C}$ can be seen as extensions or simplifications of the minimal and maximal MIDs respectively.

Example 19. Figures 4a and 4b show the minimal and the maximal MID respectively of the equivalence class including the MID of Figure 1.

Proposition 9. *The MIDs in $\mathcal{C}$ form a bounded distributive ranked lattice whose meet and join correspond respectively to the intersection and the union of the edge sets. Its maximum is the maximal MID, whilst its minimum is the minimal MID. The rank is determined by the difference between the number of edges (Y_j, Y_i), $j < i$, $i \in \mathbb{V}$, and n_{min}.*

(a) Example of a minimal MID. (b) Maximal MID in the same equivalence class of 5a.

Figure 5: Minimal and maximal MID of a simple equivalence class.

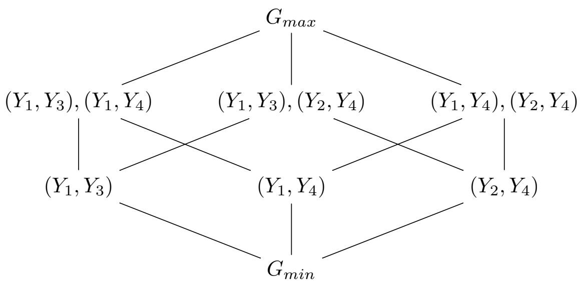


Figure 6: Hasse diagram of the equivalence class comprising the MIDs in Figure 5. The vertices labelled with edges correspond to the MID obtained by the union of those edges with the edge set of G_{min}.

Proof. This result follows directly by observing that equivalent MIDs can be thought of as a power set over the edge set of the diagrams.

The rank of G_{min} and of G_{max} are n_{min} and $n_{max} = \sum_{i \in \mathbb{V}} i - 1$, respectively and $\mathcal{C}$ has $n_{\mathcal{C}} = 2^{n_{max} - n_{min}}$ elements. Although $n_{\mathcal{C}}$ is usually very large, its elements can be ordered accordingly to a highly regular lattice, as noted in Proposition 9, thus providing some computational advantages that we outline below.

Example 20. Consider for example the equivalence class $\mathcal{C}$ comprising the MID in Figure 1. It is easy to deduce that for $\mathcal{C}$, $n_{min} = 4$, $n_{max} = 12$. This equivalence class then consists of 256 MIDs. We can further note that within this class there are $\binom{8}{i}$ MIDs having $n_{min} + i$ edges terminating into random vertices.

Analogously to lattices, we can now define the concept of a cover of one MID with respect to another.

Definition 9. Let G and G' be two MIDs in $\mathcal{C}$. We say that G' covers G if for $i \in \mathbb{V}$, $j < i$, the edge set $E(G)$ is equal to $E(G') \cup \{(Y_j, Y_i)\}$.

Example 21. Consider now the equivalence class comprising the minimal and maximal MIDs in Figure 5. Since $n_{max} - n_{min} = 3$, this equivalence class has 8 different MIDs. The Hasse diagram of this lattice is reported in Figure 6. This is a diagram whose vertices are the MIDs in the equivalence class and a line goes upward from x to y whenever y covers x, for any vertices x and y.

Theorem 2 shows how the expected utility vectors of two MIDs in the same equivalence class differ if one MID covers the other.

Theorem 2. *Let G and G' be two MIDs in $\mathcal{C}$ and let $\bar{\boldsymbol{U}}_k$ and $\bar{\boldsymbol{U}}'_k$, $k \in [n]$, be conditional utility vectors for G and G' respectively. Assume that G' covers G, that, for some $i, j \in [n]$, $E(G') = E(G) \cup \{(Y_j, Y_i)\}$ and that $\mathcal{S}(G^M) \neq \mathcal{S}(G'^M)$, i.e. the moralized versions of the MIDs have different skeletons. The dimension of the conditional expected utility vectors $\bar{\boldsymbol{U}}'_k$ is equal to $c_k r_j$ for $k = i$ and for any $k = l$, where l is the index of those vertices such that $(Y_j, Y_l) \notin E(G)$, $j < l < i$.*

Proof. Again it follows directly by identifying the sets B_i for which the two MIDs differ.

Theorem 2 implies that if two MIDs have the same skeleton, then not only the entries of their conditional expected utility vectors have the same polynomial structure, but the vectors have also the exact same dimension. Corollary 3 then shows how the expected utility vectors of G and G' differ if $G \prec G'$, where $\prec$ denotes the order relationships associated to $\mathcal{C}$.

Corollary 3. *Let G and G' be two MIDs be in $\mathcal{C}$ and let $G \prec G^{'}$. Then the difference between the expected utility vectors associated to G and $G^{'}$ computed by Algorithm 3.2 is equal to the union of the differences between any pair G_i, G_j of elements of a maximal chain between G and $G^{'}$.*

Proof. This easily follows by noting that $\mathcal{C}$ respects the Jordan-Dedeking chain condition.

Corollary 3 can also be used to deduce how the conditional expected utilities of any two MIDs, G and G' in $\mathcal{C}$ differ at any stage of their evaluations. First, the differences between G and $G \wedge G'$ are deduced from Corollary 3. Then the same procedure is iterated for $G \wedge G'$ and G'. Note that the maximal chains can be automatically identified by looking at the Hasse diagram associated to the equivalence class $\mathcal{C}$.

We do not fully explore here the computational advantages associated to the lattice structure of $\mathcal{C}$, but we present an example of how this lattice can help a DM when eliciting the edge set of an MID. From a computational point of view, Theorem 2 guarantees that the evaluation of two equivalent MIDs with the same skeleton requires the same number of operations. Therefore, if a DM is not sure whether a certain conditional independence statement holds or not, the edge associated to such statement can be simply added to the diagram, given that the resulting MID has the same skeleton.

6. Asymmetric decision problems

The MID framework represents uniform decision problems only. However, real decision problems often exhibit asymmetries of various kinds. In [31] asymmetries are 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. There are four types of functional asymmetries that can occur:

- **chance $\to$ chance**: the outcome of random variables restricts the outcomes of other random variables;
- **chance $\to$ decision**: the outcome of random variables restricts the options of controlled variables;
- **decision $\to$ chance**: decisions taken restrict the possible outcomes of random variables;
- **decision $\to$ decision**: decisions taken restrict the options of other controlled variables.

Heuristically, for each of these asymmetries the observation of $\boldsymbol{y}_A$, $A \subset [n]$, restricts the space $\boldsymbol{\mathcal{Y}}_B$ associated to a vector $\boldsymbol{Y}_B$, such that $A \cap B = \emptyset$. This new space, $\boldsymbol{\mathcal{Y}}_B'$ say, is a subspace of $\boldsymbol{\mathcal{Y}}_B$. In purely inferential settings such asymmetries are often depicted by context-specific BNs [43].

In Theorem 3 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 conditional expected utility vector. This implies that the polynomial structure of the conditional expected utility vectors and at times also their dimension change. 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 conditional expected utility vectors in more general cases can be deduced through a sequential application of Theorem 3.

Theorem 3. *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 $j < t \leq z$, $\bar{\boldsymbol{U}}_t$ has $\prod_{s \in B_t \setminus \{i \cup j\}} r_s$ rows with no monomials;*
- *for $i < t \leq j$, $\bar{\boldsymbol{U}}_t$ 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 r'_{tba} monomials of degree d_{tba}, where, for $a = x, \dots, m$ and $b = l, \dots, a$,*

$$r'_{tba} = \left(\binom{a-x}{b-l} - 1 \right) \prod_{s=t}^{j_a} r_s / r_j;$$

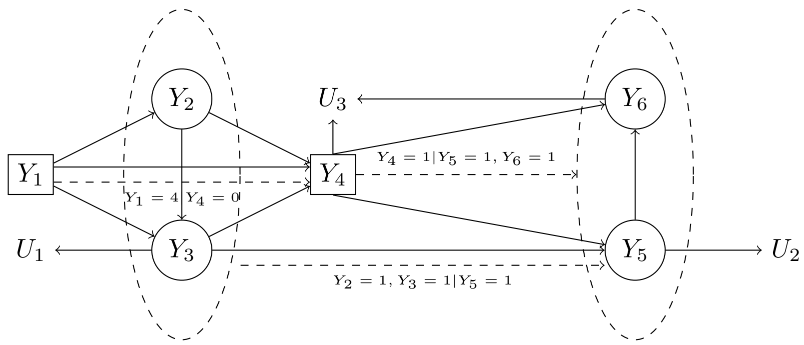


Figure 7: Representation of the asymmetric version of the MID of Figure 1 through a sequential influence diagram.

- *for $t \leq i$, each row of $\bar{\boldsymbol{U}}_t$ has in the notation of Theorem 1, r''_{tba} monomials of degree d_{tba}, where for $a = x, \dots, m$ and $b = l, \dots, a$*

$$r''_{tba} = \left(\binom{a-x}{b-l} - 1\right)\prod_{s=t}^{j_a} r_s/(r_j \cdot r_i).$$

The proof of this theorem is provided in Appendix A.6. Corollary 4 gives a characterization of simple asymmetries between any two variables, whether they are controlled or non-controlled. This follows from Theorem 3 since controlled variables can be thought of as a special case of random ones.

Corollary 4. *In the notation of Theorem 1 and under the assumptions of Theorem 3, 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 $\bar{\boldsymbol{U}}_t$ has $\prod_{s \in B_t \setminus \{i \cup j\}} r_s$ rows with no monomials;*
- *for $i < t \leq j$, $\bar{\boldsymbol{U}}_t$ 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 r'_{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 $\bar{\boldsymbol{U}}_t$ have a number of monomials of degree d_{tba} between r''_{tba} and r_{tba}, for $a = x, \dots, m$ and $b = l, \dots, a$.*

Example 22. [Example 2 continued] Assume that the DM now believes her decision problem is characterized by three composite asymmetries:

- whenever she decides to close the power source, then the population cannot be evacuated from the area: $Y_1 = 1 \Rightarrow Y_4 = 0$;
- if either the deposition or the dispersion levels in the area are high, then the human intake is high: $Y_2 = 1 \vee Y_3 = 1 \Rightarrow Y_5 = 1$;
- if the evacuation option is followed then both the human intake and the stress levels in the population are high: $Y_4 = 1 \Rightarrow Y_5 = 1 \wedge Y_6 = 1$.

A graphical representation of these asymmetries is given in Figure 7, in the form of a sequential influence diagram [27]. 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 Figure 1 graphically captures the asymmetries, most of its transparency is now lost. Instead asymmetries have the opposite effect on any polynomial representation of MIDs by greatly simplifying the structure of the conditional expected utilities.

Example 23 (Example 22 continued). In this asymmetric framework the first row of $\bar{\boldsymbol{U}}_6$ corresponds to $k_3\psi_{311}p_{6111}$, whilst its second row is empty. This is because according to Theorem 3 the monomial $k_3\psi_{301}p_{6011}$ in equation (12) 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 $\bar{\boldsymbol{U}}_5$ 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 EUMaximization suggested that $Y_4 = 0$ is optimal if $Y_3 = 1$ and that $Y_4 = 1$ is preferred if $Y_3 = 0$. The entry of $\bar{\boldsymbol{U}}_3$ for which $Y_2 = 1$ and $Y_1 = 1$ can be written as

$$\sum_{i,j=0,1}((k_2\psi_{21} + k_3\psi_{311}p_{6111}(1 + kk_2\psi_{21}))p_{5110}p_{3011} + k_1\psi_{1i}p_{3i11} + kk_1k_3\psi_{11}p_{5101}p_{3111}((1 + k_2\psi_{21})\psi_{3j0}p_{6j10})).$$

This polynomial consists of only nine monomials. This compared with the number of monomials in the symmetric case, 42 (see Table 3), means that even in this small problem we obtain a reduction of the number of monomials 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 [44]. Finally it is possible to develop a variant of Algorithm 3.2 which explicitly takes into account the asymmetries of the problem *during* its evaluation. Note that this approach would be computationally even more efficient, since this would require the computation of a smaller number of monomials/polynomials.

7. Discussion

The symbolic approach to inference in probabilistic graphical models has been extensively studied in the literature and implemented in practice using different softwares. However very few authors have looked at this approach for decision making. Here we have fully defined such a symbolic approach for MIDs and developed a complete toolkit to deal with standard operations for MIDs from a symbolic point of view, such as their evaluation, possible manipulations of the diagram and asymmetries. We have further provided an implementation of our methodology into a computer algebra system which, under the conditions we discussed above and more extensively in [23], can be simply extended to apply to much more complex problems.

The fairly recent recognition that probability models can be represented as polynomials has started a whole new area of research called algebraic statistics [45, 46]. Here techniques from, among the others, algebraic geometry and computational commutative algebra are used to gain insights into the *structure* and the properties of certain statistical models. Recently there has been significant attention on the study of probabilistic graphical models and in particular on the BN model [47, 48]. As far as we know, this paper is the first to apply these exciting new developments to the study of IDs and then to more general decision problems.

Although we mention the several potential benefits deriving from the symbolic approach we developed here, a more systematic study of how, for example, qualitative beliefs and consequently various sensitivity analyses can be performed has yet to be fully studied. Our symbolic definition could further allow DMs to simply work with the computer algebra expressions of a decision problem without defining an explicit graphical representation of the problem. If the underlying structure is very asymmetric, then our algebraic representation would still work exactly, whilst there may simply not be any associated graphical representation which is not contrived. We have some encouraging results in this direction that will be reported in future papers.

Expected utility also exhibits a similar polynomial representation in the case the variables take values in continuous spaces. In this case the unknown quantities of the polynomials are low order moments. Examples of these polynomials are presented in [23] and [49]. 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 expected utilities in a continuous domain will be reported later.

Acknowledgments

We acknowledge that Manuele Leonelli has been funded by a scholarship of the Department of Statistics of The University of Warwick, whilst James Q. Smith is partly funded by the EPRSC grant EP/K039628/1.

References

[1] A. Tversky, D. Kahneman, Judgment under uncertainty: heuristics and biases, Science 185 (4157) (1974) 1124–1131.
[2] R. A. Howard, J. E. Matheson, Influence diagrams, in: Readings on the principles and applications of decision analysis, 1983, pp. 719–763.
[3] R. A. Howard, J. E. Matheson, Influence diagrams, Decision Analysis 2 (3) (2005) 127–143.
[4] C. Bielza, M. Gómez, P. P. Shenoy, A review of representation issues and modeling challenges with influence diagrams, Omega 39 (3) (2011) 227–241.
[5] F. V. Jensen, T. D. Nielsen, Probabilistic decision graphs for optimization under uncertainty, Annals of Operations Research 204 (1) (2013) 223–248.
[6] M. Gómez, Real-world applications of influence diagrams, in: Advances in Bayesian networks, Springer, 2004, pp. 161–180.
[7] R. D. Shachter, Evaluating influence diagrams, Operations Research (6) (1986) 871–882.
[8] S. M. Olmsted, On representing and solving decision problems, Ph.D. thesis, Stanford University (1984).
[9] Y. B. Canbolat, K. Chelst, N. Garg, Combining decision tree and MAUT for selecting a country for a global manufacturing facility, Omega 35 (3) (2007) 312–325.
[10] F. Jensen, F. V. Jensen, S. L. Dittmer, From influence diagrams to junction trees, in: Proceedings of the 10th Conference on Uncertainty in Artificial Intelligence, 1994, pp. 367–373.
[11] A. L. Madsen, F. V. Jensen, Lazy evaluation of symmetric Bayesian decision problems, in: Proceedings of the 15th conference of Uncertainty in Artificial Intelligence, 1999, pp. 382–390.
[12] M. Gómez, C. Bielza, J. A. Fernández del Pozo, S. Ríos-Insua, A graphical decision-theoretic model for neonatal jaundice., Medical Decision Making: an International Journal of the Society for Medical Decision Making 27 (3) (2007) 250–265.
[13] E. Castillo, J. M. Gutierrez, A. S. Hadi, C. Solares, Symbolic propagation and sensitivity analysis in Gaussian Bayesian networks with application to damage assessment, Artificial Intelligence in Engineering 11 (2) (1997) 173–181.
[14] H. Chan, A. Darwiche, Sensitivity analysis in Bayesian networks: from single to multiple parameters, in: Proceedings of the 20th conference on Uncertainty in Artificial Intelligence, 2004, pp. 67–75.
[15] D. Bhattacharjya, R. D. Shachter, Sensitivity analysis in decision circuits, in: Proceedings of the 24th Conference on Uncertainty in Artificial Intelligence, 2008, pp. 34–42.
[16] D. Bhattacharjya, R. D. Shachter, Three new sensitivity analysis methods for influence diagrams, in: Proceedings of the 26th Conference on Uncertainty in Artificial Intelligence, 2010, pp. 56–64.
[17] E. Castillo, J. M. Gutierrez, A. S. Hadi, Goal oriented symbolic propagation in Bayesian networks, in: Proceedings of the 13th National Conference on Artificial Intelligence, Vol. 1, 1996, pp. 1263–1268.
[18] A. Darwiche, A differential approach to inference in Bayesian networks, Journal of the ACM 50 (3) (2003) 280–305.
[19] D. Bhattacharjya, R. D. Shachter, Evaluating influence diagrams with decision circuits, in: Proceedings of the 23rd Conference on Uncertainty in Artificial Intelligence, 2007, pp. 9–16.
[20] R. D. Shachter, D. Bhattacharjya, Dynamic programming in influence diagrams with decision circuits, in: Proceedings of the 26th Conference on Uncertainty in Artificial Intelligence, 2010, pp. 509–516.
[21] E. Castillo, J. M. Gutierrez, A. S. Hadi, Parametric structure of probabilities in Bayesian networks, in: Symbolic and Quantitative Approaches to Reasoning and Uncertainty, 1995, pp. 89–98.
[22] R. L. Keeney, H. Raiffa, Decision with multiple objectives: preferences and value trade-offs, Cambridge University Press, 1976.
[23] M. Leonelli, J. Q. Smith, Bayesian decision support for complex systems with many distributed experts, invited revision to the Annals of Operations Research (2015).
[24] J. E. Smith, S. Holtzman, J. E. Matheson, Structuring conditional relationships in influence diagrams, Operations Research 41 (2) (1993) 280–297.
[25] R. Demirer, P. P. Shenoy, Sequential valuation networks for asymmetric decision problems, European Journal of Operational Research 169 (1) (2006) 286–309.

[26] T. D. Nielsen, F. V. Jensen, Representing and solving asymmetric decision problems, International Journal of Information Technology & Decision Making 2 (02) (2003) 217–263.
[27] F. V. Jensen, T. D. Nielsen, P. P. Shenoy, Sequential influence diagrams: a unified asymmetry framework, International Journal of Approximate Reasoning 42 (1) (2006) 101–118.
[28] F. V. Jensen, M. Vomlelová, Unconstrained influence diagrams, in: Proceedings of the 18th conference on Artificial Intelligence, 2002, pp. 234–241.
[29] D. Bhattacharjya, R. D. Shachter, Formulating asymmetric decision problems as decision circuits, Decision Analysis 9 (2) (2012) 138–145.
[30] R. G. Cowell, P. A. Thwaites, J. Q. Smith, Decision making with decision event graphs, Tech. Rep. 10-15, CRISM, The University of Warwick (2010).
[31] F. V. Jensen, T. D. Nielsen, Bayesian networks and decision graphs, Springer, 2009.
[32] J. Q. Smith, Bayesian decision analysis: principles and practice, Cambridge University Press, 2010.
[33] A. P. Dawid, P. Constantinou, A formal treatment of sequential ignorability, Statistics in Biosciences 6 (2) (2014) 166–188.
[34] R. L. Keeney, Multiplicative utility functions, Operations Research 22 (1) (1974) 22–34.
[35] D. Von Winterfeldt, G. W. Fischer, Multi-attribute utility theory: models and assessment procedures, Springer, 1975.
[36] R. G. Cowell, A. P. Dawid, S. L. Lauritzen, D. J. Spiegelhalter, Probabilistic Networks and Expert Systems, Springer-Verlag, 1999.
[37] J. Q. Smith, Influence diagrams for statistical modelling, The Annals of Statistics 17 (2) (1989) 654 – 672.
[38] D. A. Cox, J. Little, D. O'Shea, Ideals, varieties, and algorithms: an introduction to computational algebraic geometry and commutative algebra, Springer, 2007.
[39] G. F. Cooper, The computational complexity of probabilistic inference using Bayesian belief networks, Artificial Intelligence 42 (2-3) (1990) 393–405.
[40] P. A. Parrilo, Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization, Ph.D. thesis, California Institute of Technology (2000).
[41] J. Q. Smith, Influence diagrams for Bayesian decision analysis, European Journal of Operational Research 40 (3) (1989) 363–376.
[42] S. L. Lauritzen, Graphical Models, Oxford University Press, 1996.
[43] C. Boutilier, N. Friedman, M. Goldszmidt, D. Koller, Context-specific independence in Bayesian networks, in: Proceedings of the 12th Conference on Uncertainty in Artificial Intelligence, 1996, pp. 115–123.
[44] C. Görgen, M. Leonelli, J. Q. Smith, A differential approach for staged trees, accepted to ECSQARU2015 (2015).
[45] E. Riccomagno, A short history of algebraic statistics, Metrika 69 (2-3) (2009) 397–418.
[46] P. Gibilisco, E. Riccomagno, M. P. Rogantin, H. P. Wynn, Algebraic and geometric methods in statistics, Cambridge University Press, 2010.
[47] M. Drton, B. Sturmfels, S. Sullivant, Lectures on algebraic statistics, Springer, 2009.
[48] L. D. Garcia, M. Stillman, B. Sturmfels, Algebraic geometry of Bayesian networks, Journal of Symbolic Computation 39 (3-4) (2005) 331–355.
[49] M. Leonelli, J. Q. Smith, Dynamic Uncertainty Handling for Coherent Decision Making in Nuclear Emergency Response, in: Risk Management for Complex Socio-Technical Systems, 2013, to appear.

Appendix A. Proofs

Appendix 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 $i \in [n]$,

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

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 DM's preferences are a function of Y_n only through $k_m U_m(\boldsymbol{y}_{P_m})$, since by construction $n = j_m \in \mathbb{J}$. Therefore this quantity can be either maximized or marginalized as in equation (3) to compute $\bar{U}_n(\boldsymbol{y}_{B_n})$. 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

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

Now consider Y_{n-1}. If $n-1 \notin \mathbb{J}$, then $\bar{U}^n$ is a function of Y_{n-1} only through $\bar{U}_n$. Therefore maximization and marginalization steps can be computed as in equation (6) to compute $\bar{U}_{n-1}(\boldsymbol{y}_{B_{n-1}})$. 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

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

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

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

where

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

Therefore optimization and marginalization steps can be performed over U'_{m-1} as specified in equations (4) and (5) respectively. Then note that $\bar{u}^{n-1}$ can be written as

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

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

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

Therefore at the following step, when considering Y_{j-1}, we can proceed as done with Y_{n-1} by maximization and marginalization in equations (4)-(6) to compute $\bar{U}^{j-1}$. Thus at the conclusion of the procedure, $\bar{U}_1$ yields the expected utility of the optimal decision.

Appendix 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 $C_I^i = \{z \in \mathbb{V} : i \leq z \leq j_I\}$. The conditional expected utility function of equations (3)-(6) can be (less intuitively) written as

$$\bar{U}_i(\boldsymbol{y}_{B_i}) = \sum_{I \in \mathcal{P}_0(\{l, \dots, m\})} \bar{U}_i^I(\boldsymbol{y}_{B_i}) = \sum_{I \in \mathcal{P}_0(\{l, \dots, m\})} k^{n_I - 1} \prod_{s \in I} k_s U_s(\boldsymbol{y}_{P_s}) \sum_{\boldsymbol{y}_{C_I^i} \in \boldsymbol{\mathcal{Y}}_{C_I^i}} P(\boldsymbol{y}_{C_I^i} | \boldsymbol{y}_{B_i}), \tag{A.1}$$

where

$$P(\boldsymbol{y}_{C_I^i} | \boldsymbol{y}_{B_i}) = \prod_{t \in C_I^i} P(y_t | \boldsymbol{y}_{B_t}). \tag{A.2}$$

The conditional expected utility 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 $U_{\max_I} = U_{\max_J}$, $\bar{U}_i^I(\boldsymbol{y}_{B_i})$ and $\bar{U}_i^J(\boldsymbol{y}_{B_i})$ have the same polynomial structure since $C_I^i = C_J^i$. Now for $a = l, \dots, m$ and $b = l, \dots, a$, the binomial coefficient $\binom{a-l}{b-l}$ 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 equation (8) counts the correct number of monomials having a certain degree since $\boldsymbol{\mathcal{Y}}_{C_I(i)} = \times_{t\in C_I^i}\mathcal{Y}_t$. Further note that considering each combination of b and a in the ranges specified above, we consider each element of $\mathcal{P}_0(\{l,\dots,m\})$.

By having a closer look at d_{iba} in equation (8) 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 k, 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 $U_s(\boldsymbol{y}_{P_s})$ for $s \in C_I^i$. The last term w_{ia} corresponds to the degree deriving from the probabilistic part of equation (A.1), which is equal to the number of non-controlled vertices between Y_i and $Y_{j_{\max_I}}$ (both included) as shown by equation (A.2).

Since the set B_i includes the arguments of $\bar{U}_i(\boldsymbol{y}_{B_i})$ and $\boldsymbol{\mathcal{Y}} = \times_{i\in[n]}\mathcal{Y}_i$, equation (7) guarantees that the dimension of the conditional expected utility vector is $\prod_{t\in B_i} r_t$.

Appendix 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 $\Pi_j' = \{\Pi_j \setminus i \cup \Pi_i\}$ and $\Pi_i' = \{j \cup \Pi_i \cup \Pi_j \setminus i\}$. Call $\Pi_j^i = \{\Pi_j \setminus i\}$. It then follows that

$$\begin{aligned} p_{iy_i\pi_i'} &= P(y_i \mid \boldsymbol{y}_{\Pi_i'}) = P(y_i \mid \boldsymbol{y}_{\Pi_j^i}, \boldsymbol{y}_{\Pi_i}, y_j) = \frac{P(y_j \mid \boldsymbol{y}_{\Pi_j^i}, \boldsymbol{y}_{\Pi_i}, y_i)P(y_i \mid \boldsymbol{y}_{\Pi_j^i}, \boldsymbol{y}_{\Pi_i})}{P(y_j \mid \boldsymbol{y}_{\Pi_j^i}, \boldsymbol{y}_{\Pi_i})} \\ &= \frac{P(y_j \mid \boldsymbol{y}_{\Pi_j})P(y_i \mid \boldsymbol{y}_{\Pi_i})}{P(y_j \mid \boldsymbol{y}_{\Pi_j^i}, \boldsymbol{y}_{\Pi_i})} = \frac{P(y_j \mid \boldsymbol{y}_{\Pi_j})P(y_i \mid \boldsymbol{y}_{\Pi_i})}{\sum_{y_i\in\mathcal{Y}_i} P(y_j \mid y_i, \boldsymbol{y}_{\Pi_j^i})P(y_i \mid \boldsymbol{y}_{\Pi_i})} = \frac{p_{jy_j\pi_j}p_{iy_i\pi_i}}{\sum_{y_i\in\mathcal{Y}_i} p_{jy_j\pi_j}p_{iy_i\pi_i}}, \end{aligned}$$

and

$$p'_{jy_j\pi_j'} = P(y_j \mid \boldsymbol{y}_{\Pi_j'}) = P(y_j \mid \boldsymbol{y}_{\Pi_j^i}, \boldsymbol{y}_{\Pi_i}) = \sum_{y_i\in\mathcal{Y}_i} P(y_j \mid \boldsymbol{y}_{\Pi_j})P(y_i \mid \boldsymbol{y}_{\Pi_i}) = \sum_{y_i\in\mathcal{Y}_i} p_{jy_j\pi_j}p_{iy_i\pi_i}.$$

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

Appendix A.4. Proof of Lemma 1.

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 $\bar{\boldsymbol{U}}_{i+1}$ and $\boldsymbol{p}_j$. 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 3.2 simply computes an EUMarginalization between $\bar{\boldsymbol{U}}_{j+1}$ and $\boldsymbol{p}_i$ instead of $\boldsymbol{p}_j$. Therefore the entries of $\bar{\boldsymbol{U}}_j$ have $r_{jba}' = r_i r_{(j+1)ba}/r_j$ monomials of degree $d_{(j+1)ba}$ and, until the adjusted algorithm computes $\bar{\boldsymbol{U}}_i$, 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 $\bar{\boldsymbol{U}}_i$. 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 $\bar{\boldsymbol{U}}_k$ 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 3.2 now performs one EUMarginalization less and, from Proposition 2, we deduce that $\bar{U}_k'$ has now r_{kba}/r_i monomials of degree $d_{kba} - 1$.

Appendix A.5. Proof of Proposition 6.

If Y_i is father of Y_k we have that

$$p'_{ky_k\pi'_k} = P(y_k \mid \boldsymbol{y}_{\Pi'_k}) = P(y_k \mid \boldsymbol{y}_{\Pi_i}, \boldsymbol{y}_{\Pi_k^i}) = \sum_{y_i \in \mathcal{Y}_i} P(y_k \mid \boldsymbol{y}_{\Pi_k}, \boldsymbol{y}_{\Pi_i}) P(y_i \mid \boldsymbol{y}_{\Pi_k^i}, \boldsymbol{y}_{\Pi_i}) \tag{A.3}$$
$$= \sum_{y_i \in \mathcal{Y}_i} P(y_k \mid \boldsymbol{y}_{\Pi_k}) P(y_i \mid \boldsymbol{y}_{\Pi_i}) = \sum_{y_i \in \mathcal{Y}_i} p_{ky_k\pi_k} p_{iy_i\pi_i}$$

If Y_i is a parent but not the father of Y_j, then $P(y_i \mid \boldsymbol{y}_{\Pi_j^i}, \boldsymbol{y}_{\Pi_i})$ as in Equation (A.3) can be written as

$$P(y_i \mid \boldsymbol{y}_{\Pi_j^i}, \boldsymbol{y}_{\Pi_i}) = P(y_i \mid \boldsymbol{y}_{\Pi_j^{>i}}, \boldsymbol{y}_{\Pi_j^{<i}}, \boldsymbol{y}_{\Pi_i}) = \frac{P(\boldsymbol{y}_{\Pi_j^{>i}} \mid y_i, \boldsymbol{y}_{\Pi_j^{<i}}, \boldsymbol{y}_{\Pi_i}) P(y_i \mid \boldsymbol{y}_{\Pi_i})}{\sum_{y_i \in \mathcal{Y}_i} P(\boldsymbol{y}_{\Pi_j^{>i}} \mid y_i, \boldsymbol{y}_{\Pi_j^{<i}}, \boldsymbol{y}_{\Pi_i}) P(y_i \mid \boldsymbol{y}_{\Pi_i})}$$
$$= \frac{\prod_{l \in \Pi_j^{>i}} \sum_{y_{\Pi_i^{j,l}} \in \mathcal{Y}_{\Pi_i^{j,l}}} P(y_l \mid \boldsymbol{y}_{\Pi_l}) P(y_i \mid \boldsymbol{y}_{\Pi_i})}{\sum_{y_i \in \mathcal{Y}_i} \prod_{l \in \Pi_j^{>i}} \sum_{y_{\Pi_i^{j,l}} \in \mathcal{Y}_{\Pi_i^{j,l}}} P(y_l \mid \boldsymbol{y}_{\Pi_l}) P(y_i \mid \boldsymbol{y}_{\Pi_i})},$$

where $\Pi_k^{<i} = \{\Pi_k \setminus \{i, \dots, k-1\}\}$.

Appendix A.6. Proof of Theorem 3

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 $y_k \in \mathcal{Y}_k$, $\pi_k \in \boldsymbol{\mathcal{Y}}_{\Pi_k}$, $y_l \in \mathcal{Y}_l$, $\pi_l \in \boldsymbol{\mathcal{Y}}_{\Pi_l}$, $\pi_t \in \boldsymbol{\mathcal{Y}}_{P_t}$ and $\pi_s \in \boldsymbol{\mathcal{Y}}_{P_s}$. Thus these monomials have to be set equal to zero.

For $j < t \leq z$, $\bar{\boldsymbol{U}}_t$ 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 $\boldsymbol{\psi}_{xP_x}$ 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 $\bar{\boldsymbol{U}}_t$, for $a = x, \dots, m$ and $b = l, \dots, a$ (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 $\bar{\boldsymbol{U}}_t$. 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}, $a = x, \dots, m$, $b = l, \dots, a$, implying the event $Y_i = y_i \wedge Y_j = y_j$.

Appendix B. Maple$^{\text{TH}}$ Code

```
### Computation of the highest index in each parent set of a utility node ###
# Inputs: PiU::table, parent sets of utility nodes; m::integer, number of 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 expected utility 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, expected utility 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:

### EUDuplication of a utility vector and an expected utility 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, psiprime, psitemp, utemp, x, sx, y, h, z:
i, uprime, psiprime, psitemp, utemp:= max(PiU[j]), [], [], 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, Bi[i+1] union PiU[j]) then sx := sx*r[y] end if
    end do:
    if member(x, Bi[i+1]) then for l to Size(psitemp)[2]/sx do for z to r[x] do
          psiprime := [op(psiprime), op(convert(psitemp, list)[(l-1)*sx+1 .. l*sx])]
        end do end do:
      psitemp, psiprime := psiprime, []:
    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(utemp, list)[(l-1)*sx+1 .. l*sx])]
      end do end do:
      utemp,  uprime := uprime, []:
end if end if end do:
return utemp, psitemp:
end proc:

### EuMultiSum between an expected utility 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, expected utility vector after an EUMultiSum
```

```
EUMultiSum := proc(u, psi, j, PiV, PiU, r, Bi, J) local i, uprime, psiprime, ut;
i := max(PiU[j]);
uprime, psiprime := EUDuplicationPsi(u, psi, j, PiV, PiU, r, Bi, J);
uprime := convert(uprime, list);
if Size(uprime)[1] = 0 then  ut := k[j]*~psiprime
else ut := h*~k[j]*~psiprime*~uprime +~ uprime +~ k[j]*~psiprime end if;
return ut:
end proc:

### EUDuplication of a probability vector and an expected utility 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:
uprime, pprime, ptemp, utemp   := [], [],  p[i], u[i+1]:
Uni:= Bi[i+1] union PiV[i] union PiU[j];
if member(i, J) then member(i, J, '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, Bi[i+1] union PiU[j]) then
        for l to Size(ptemp)[2]/sx do for z to r[x] do
            pprime := [op(pprime), op(convert(ptemp, list)[(l-1)*sx+1 .. l*sx])]
        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(utemp, list)[(l-1)*sx+1 .. l*sx])]
        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(ptemp, list)[(l-1)*sx+1 .. l*sx])]
        end do end do;
        ptemp, pprime := psiprime, [];
      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(utemp, list)[(l-1)*sx+1 .. l*sx])]
        end do end do;
        utemp, uprime := uprime, []:
end if end if end do end if:
```

```
return utemp, ptemp:
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, expected utility vector after EUMarginalization
EUMarginalization := proc(u, p, i, PiV, PiU, r, Bi, J)
local uprime, pprime, rows, cols, l, one, zero, row, Iprime, ut;
uprime, pprime := EUDuplicationP(u, p, i, PiV, PiU, r, Bi, J);
pprime, rows, cols := convert(pprime, list), 1,  Size(pprime)[2];
for l to i-1 do if member(l, PiV[i]) then rows := rows*r[l] end if end do;
one, zero := convert(Vector(r[i], 1), list),  convert(Vector(cols, 0), list):
row, Iprime := [op(zero), op(one)],  [op(one)]:
for l to rows-1 do Iprime := [op(Iprime), op(row)] end do;
Iprime := Matrix(rows, cols, [op(Iprime), op(one)]);
ut := convert(Multiply(Iprime, Vector[column](uprime*~pprime)), list);
return ut:
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:

### EUMaximization over a decision space ###
# Inputs: u::table; i::integer; r::table
# Outputs: u[i]::list, expected utility vector after EUMaximization
EUMaximization := proc(u, i, r) local opt, Istar, j, l, zero;
opt := Maximize(r, i, Size(u[i+1])[2]/r[i]);
zero := convert(Vector(Size(u[i+1])[2], 0), list);
Istar := [];
for j to Size(u[i+1])[2]/r[i] do for l to r[i] do
    if opt[j] = l then  Istar := [op(Istar), 1]  else Istar := [op(Istar), 0] end if
 end do;
  if j < Size(u[i+1])[2]/r[i] then Istar := [op(Istar), op(zero)] end if end do;
Istar := Matrix(Size(u[i+1])[2]/r[i], Size(u[i+1])[2], Istar);
u[i] := convert(Multiply(Istar, Vector[column](u[i+1])), list);
return u[i]:
end proc:

### 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, expected utility vectors;
```

```
SymbolicExpectedUtility := proc(p, psi, PiV, PiU, n, m, De, V, r)
 local J, Bi, utemp, i, j, ceu,best;
with(LinearAlgebra): with(ArrayTools):
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:
```

Consider the MID in Figure 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 expected utility 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]
```


Paper No. 15-07, www.warwick.ac.uk/go/crism