Available online at www.sciencedirect.com

ScienceDirect

Artificial Intelligence 172 (2008) 1018–1044

Artificial Intelligence

www.elsevier.com/locate/artint

Complexity results and algorithms for possibilistic influence diagrams

Laurent Garcia [a], Régis Sabbadin [b,*]

[a] *LERIA, University of Angers, France*
[b] *INRA-BIA Toulouse, France*

Received 30 November 2006; received in revised form 27 November 2007; accepted 27 November 2007

Available online 8 December 2007

Abstract

In this article we present the framework of *Possibilistic Influence Diagrams* (PID), which allows to model in a compact form problems of sequential decision making under uncertainty, when only ordinal data on transitions likelihood or preferences are available. The graphical part of a PID is exactly the same as that of usual influence diagrams, however the semantics differ. Transition likelihoods are expressed as possibility distributions and rewards are here considered as satisfaction degrees. Expected utility is then replaced by anyone of the two possibilistic qualitative utility criteria (optimistic and pessimistic) for evaluating strategies in a PID. We then describe decision tree-based methods for evaluating PID and computing optimal strategies and we study the computational complexity of PID optimisation problems for both cases. Finally, we propose a dedicated *variable elimination* algorithm that can be applied to both optimistic and pessimistic cases for solving PID.

Keywords: Decision theory; Possibility theory; Causal networks; Influence diagrams

1. Introduction

For several years, there has been a growing interest in the Artificial Intelligence community towards the foundations and computational methods of decision making under uncertainty. This is especially relevant for applications to planning under uncertainty, where a suitable *strategy* (i.e. either a sequence of unconditional decisions or a decision tree) is to be found, starting from a description of the initial world, of the available decisions and their (perhaps uncertain) effects, and of the goals to reach.

Several authors have thus proposed to integrate some parts of decision theory into the paradigm of planning under uncertainty (see for example [2]). A classical model based on decision theory for planning under uncertainty is the *Markov decision processes* framework (MDP) [20], where uncertain effects of actions are represented by probability distributions and the satisfaction of agents is expressed by a numerical, additive utility function. However, transition probabilities are not always available for representing the effects of actions, especially in Artificial Intelligence applications where uncertainty is often ordinal, qualitative. The same remark applies to utilities: it is often more adequate

* Corresponding author.
E-mail addresses: laurent.garcia@info.univ-angers.fr (L. Garcia), sabbadin@toulouse.inra.fr (R. Sabbadin).

0004-3702/$ – see front matter
doi:10.1016/j.artint.2007.11.008

to represent preference over states simply with an ordering relation rather than with additive utilities. Several authors have advocated this qualitative view of decision making and have proposed qualitative versions of decision theory. [7] proposes a qualitative utility theory based on possibility theory, where preferences and uncertainty are both expressed qualitatively on the same totally ordered scale. [9,22] have extended the use of these criteria to sequential decision and have proposed *possibilistic* counterparts of the classical dynamic programming solution methods for MDP.

The MDP framework (be it stochastic or possibilistic) makes the assumption that the possible states of the world and the available actions are described *in extension*, as are transition probabilities and utility functions. However, Artificial Intelligence has a long history of focusing on *structured* models of knowledge representation. These models can be based on logic, *Constraint Satisfaction Problems* [13] or *graphical representation* (for example, *Bayesian networks* [15]). Then, naturally, models and algorithms for solving structured decision making under (stochastic) uncertainty problems were proposed, based on graphical models [11] or on logic [3].

The work described in this article builds a bridge between qualitative possibilistic decision criteria and graphical models: it incorporates the semantics of qualitative possibilistic decision theory with the graphical syntax of influence diagrams. In addition, an in-depth analysis of the resulting framework is provided, both from a complexity-theoretic point of view and from an algorithmic point of view. Such a bridge is important, especially for building well-founded decision support systems (DSS). Indeed, in a DSS, knowledge and preferences have to be elicitated from a human user and automatically manipulated so as to provide the user with well-argumented decision suggestions. Such decision problems are likely to involve many (interacting) state and decision variables, and knowledge and preferences are likely to be only "roughly" specified, by means of orderings of plausible values of state variables and preferred values of objective variables. Our work allows to deal with knowledge and preferences expressed in such a way and to compute strategies which are formally justified in the framework of possibilistic utility theory.

We should point out the relationship between the work we present and the parallel work of Pralet et al. [16,17] on the *Plausibility-Feasibility-Utility* (PFU) framework, a generic language allowing to represent knowledge and preferences in various ways and to manipulate them. The PFU framework allows to model various problems (from weighted CSP to influence diagrams) and proposes generic solution algorithms for solving them. The authors recently claimed that possibilistic (finite-horizon) MDP and possibilistic influence diagrams (PID) could be embedded in their framework. It would certainly be interesting in the future to compare the specialisation of their generic algorithm to PID with our own algorithms.

The present article is an extended version of [10]. After having presented possibilistic counterparts of expected utility (Section 2), we present (Section 3) the framework of *Possibilistic Influence Diagrams* (PID). The graphical part of a PID is exactly the same as that of an usual influence diagram, however the semantics differ. Transition likelihoods are expressed by possibility distributions, and rewards are here considered as satisfaction degrees attached to partial goals. Expected utility is then replaced by anyone of the two possibilistic qualitative utility criteria proposed by [7] for evaluating strategies in a PID. Then, we describe a decision tree representation of the PID and a decision tree-based method for evaluating it and computing an optimal strategy. In Section 4, we show some complexity results about several PID-related problems. These results enlighten the fact that optimistic optimisation problems are easier to solve than pessimistic ones. Finally (Section 5), we propose *variable elimination* algorithms for solving PID in both optimistic and pessimistic cases.

2. Possibilistic counterparts of expected utility

In a problem of (sequential) decision under uncertainty, the *state of the world*, possibly ill-known, can be represented by a combination $x = \{x_1, \ldots, x_n\}$ of values taken by a set of state variables $\mathcal{X} = \{X_1, \ldots, X_n\}$.[1] In the same way, a *decision* can be represented by a combination $d = \{d_1, \ldots, d_p\}$, chosen by a decision maker, of values of the decision variables $\mathcal{D} = \{D_1 \ldots D_p\}$, distinct from $\mathcal{X}$.

The decision maker's *strategy* expresses the way the values of the decision variables are chosen, with respect to the values of the state variables which are observed before the choice. Formally, a strategy is represented by a function $\delta : \mathcal{X} \to \mathcal{D}$, which associates a decision $d = \delta(x)$ to every possible states of the world. A strategy can also be defined

[1] X_i will denote both the variable name and its domain, the meaning should be clear from the context. In the same way, $\mathcal{X}_A$, with $A \subseteq \{1, \ldots, n\}$ will represent both a set of variables $\{X_i\}_{i \in A}$ and the Cartesian product of their domains $\bigotimes_{i \in A} X_i$. The same holds for decision variables.

by the use of a set of partial functions, $\{\delta_1, \ldots, \delta_p\}$, $\delta_j : X_{j_1} \times \cdots \times X_{jk_j} \to D_j$ which define the value taken by the decision variable D_j with respect to the values taken by a subset of the state variables (and not necessarily all of them). These partial functions allow to take into account the "sequential" aspect of decision problem: when d_1 has to be chosen, only the values of some state variables are known, then new state variables are revealed and d_2 is chosen, etc. In other words, if the ordering of decision variables is considered as a "temporal" ordering, and if the set of state variables is partitioned "accordingly", the strategy δ can be considered as "dynamic".

In a problem of decision under uncertainty defined as above, the uncertainty weighs only on the values of the state variables. The values of the decision variables are chosen deterministically with respect to the state variables, by a fixed strategy δ. In the setting of possibilistic qualitative decision theory proposed by [7], the uncertainty of the agent about the effects of strategy δ is represented by a possibility distribution π_δ mapping the set $\mathcal{X}$ of state variables values into a bounded, linearly ordered (qualitative) valuation set L equipped with an order-reversing map n. Let 1_L and 0_L denote the top and bottom elements of L respectively, $n(0_L) = 1_L$ and $n(1_L) = 0_L$. The quantity $\pi_\delta(x)$ thus represents the degree of possibility of the state x to be the consequence of strategy δ. For instance, $\pi_\delta(x) = 1_L$ means that x is completely plausible, whereas $\pi_\delta(x) = 0_L$ means that it is completely impossible.

The decision maker's preferences are expressed on pairs (x, d) and represented by an ordinal utility function $\mu : \mathcal{X} \times \mathcal{D} \to L$. $\mu(x, d)$ expresses to what extent the instantiation d of the decision variables is satisfactory when the real world is x. Let us note that, to a utility function μ and a fixed strategy δ corresponds a utility function $\mu_\delta : \mathcal{X} \to L$, defined from μ by $\mu_\delta(x) = \mu(x, \delta(x))$, $\forall x \in \mathcal{X}$. $\mu_\delta(x) = 1_L$ means that x is completely satisfactory, whereas if $\mu_\delta(x) = 0_L$, it is totally unsatisfactory. Notice that π_δ is normalised (there shall be at least one completely possible state of the world), but μ_δ may not be (it can be that no consequence is totally satisfactory).

From π_δ and μ_δ, [7] proposed two qualitative decision criteria for assessing the value of a strategy δ:

$$u^*(\delta) = \max_{x \in X} \min\bigl\{\pi_\delta(x), \mu_\delta(x)\bigr\}, \tag{1}$$

$$u_*(\delta) = \min_{x \in X} \max\bigl\{n\bigl(\pi_\delta(x)\bigr), \mu_\delta(x)\bigr\}. \tag{2}$$

u^* can be seen as an extension of the *maximax* criterion which assigns to an action the utility of its best possible consequence. On the other hand, u_* is an extension of the *maximin* criterion which corresponds to the utility of the worst possible consequence. u_* measures to what extent every plausible consequence are satisfactory. u^* corresponds to an adventurous (optimistic) attitude in front of uncertainty, whereas u_* is conservative (pessimistic).

Let us focus here on the fact that the uncertainty and the preference degrees are explicitly assumed to be measured on the same scale L. This amounts to make an assumption of *commensurability* between uncertainty and preference. This assumption does not seem very natural at first sight. However, it can be totally justified in a decision theory setting since it is a rather direct consequence of the following axiom of *comparability of strategies*:

From two strategies δ, δ', either one is preferred to the other, or the two ones are equivalent from the decision maker's point of view.

We do not develop this argument here but the interested reader is invited to refer to [8] where this argument is developed in the "non-sequential" decision theoretic setting. Furthermore, let us note that this assumption of commensurability is not related to the possibilistic approach. It is also present in the setting of expected utility theory. In the latter, preferences between strategies are invariant with respect to any positive affine transformation of the utility function, but a non-affine transformation of the utility function might change preferences between strategies.

Some authors have tried to relax the assumption of total comparability in the possibilistic setting [4]. This lead to decision procedures which are not "decisive" in the setting of (non-sequential) qualitative decision making. Indeed, if the assumption of commensurability is given up, the uncertainty measure defines an ordered set of subsets of possible states and two strategies δ and δ' will not be generally *comparable*, unless one totally dominates the other in a subset of states and they are *equivalent* in all more plausible states.

In the following, we define *Possibilistic Influence Diagrams* (PID) which allow to represent in a compact form the functions π_δ and μ_δ taking the *local* dependencies between state and decision variables into account. Then, we will focus on the *computation of strategies* which optimise one of the two criteria u_* or u^*. We will show the theoretical complexity of this computation and we will describe *exact* algorithms for the computation of strategies, inspired by *variables elimination* algorithms, used to solve problems of decision making under uncertainty represented by classical influence diagrams [11].

3. Possibilistic influence diagrams (PID)

The uncertainty π_δ and the utility μ_δ attached to a strategy δ are functions mapping $\mathcal{X}$ into L. The space needed for representing π_δ and μ_δ is a priori exponential in the number of (state and decision) variables. Using two tables to represent these functions can thus be impossible in practice. That's why, as in classical *influence diagrams* (ID) [11], they are expressed when possible in terms of *local* functions. More precisely, π_δ is generally expressed by the means of a set of conditional possibility tables $\{\pi_{X_i}(X_i|Par(X_i))\}_{X_i \in \mathcal{X}_i}$, where $Par(X_i) \subseteq \mathcal{X} \cup \mathcal{D}$ is the set of variables on which X_i depends. Let us note that each element of a possibility table belongs to L. $\pi_\delta(x)$ can then be computed if necessary from conditional possibility tables and δ, using the *possibilistic chain rule* which will be recalled in Section 3.2. In the same way, μ_δ is represented by local utility functions $\{\mu_k\}_{k=1,\dots,q}$, each one depending on a reduced number of state and decision variables.

The framework of *possibilistic influence diagrams* (PID) includes a *syntactic* part which allows to represent dependencies between variables together with an ordering of decision variables reflecting the order in which they are chosen. It also includes a *semantic* part which explicitly specifies the values of the a priori and conditional possibility tables and of the local utility functions.

3.1. Structural description of a PID

A *possibilistic influence diagram* is graphically represented by a *directed acyclic graph* (DAG) containing three different kinds of nodes:

- *chance nodes*, drawn as circles, represent state variables $X_i \in \mathcal{X} = \{X_1, \dots, X_n\}$, as in the *Bayesian Networks* (BN) framework [15].
- *decision nodes*, drawn as rectangles, represent decision variables $D_j \in \mathcal{D} = \{D_1, \dots, D_p\}$.
- *utility nodes* $\mathcal{V} = \{V_1, \dots, V_q\}$, drawn as diamonds, represent local "satisfaction degree" functions $\mu_k \in \{\mu_1, \dots, \mu_q\}$.

The local dependencies between the different variables are represented by the directed edges of the DAG. The meaning of these edges is the following:

- The edges directed toward a node X_i representing a state variable, specify the variables $Par(X_i) \subseteq \mathcal{X} \cup \mathcal{D}$ on which the conditional possibility degree of X_i depends, via $\pi_{X_i}(X_i|Par(X_i))$.
- The edges directed toward a node D_j representing a decision variable, specify the variables $Par(D_j) \subseteq \mathcal{X}$ whose values are revealed immediately before the value of D_j is chosen.
- Finally, the edges directed to a utility node V_k specify the variables $Par(V_k) \subseteq \mathcal{X} \cup \mathcal{D}$ on which the local utility function μ_k depends.

Three assumptions on the DAG structure allow to partially define the order in which state variables are revealed and decisions are taken. They also allow to define the domain of partial strategies.

(1) Decision variables are supposed to be totally ordered, according to an a priori, fixed, ordering (this ordering should be consistent with any existing oriented path between decision nodes of the DAG).
(2) The values of the state variables which are revealed are never "forgotten".
(3) Finally, in the DAG attached to a PID, utility nodes are supposed to have no successors.

Given assumptions (1) and (2), all edges in the DAG linking two decision nodes can be suppressed. This is why it is usually assumed that edges directed toward a decision node are only issued from chance nodes.

The next example shows a DAG attached to a PID:

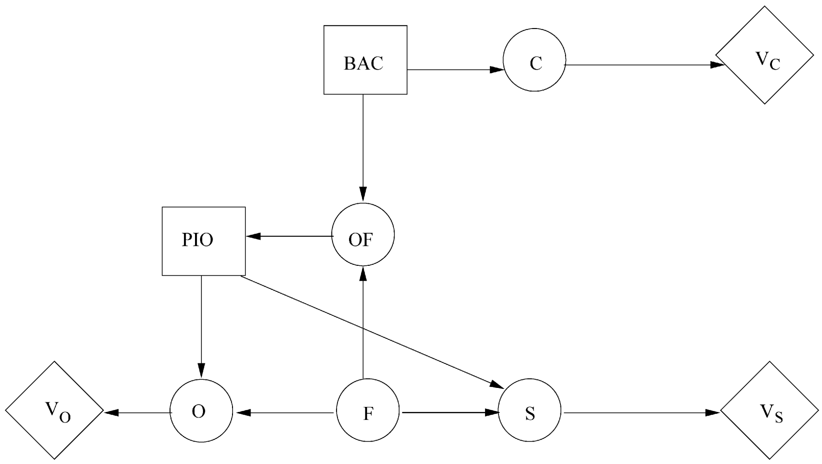

Fig. 1. An "omelette" example.

Example 1. Consider the example of Fig. 1. The problem is to prepare a six-egg omelette from a five-egg one. The new egg can be fresh or rotten.

There are two binary decision variables:

- *BAC*: break the egg apart in a cup (*BAC* = *yes* if the egg is broken apart in a cup for inspection and *BAC* = *no* if not),
- *PIO*: put it in the omelette (*PIO* = *yes* or *no*).

The state variables are the following:

- C: whether we have a cup to wash ($C = true$) or not ($C = false$).
- *OF*: whether we observe that the egg is fresh ($OF = t$) or rotten ($OF = f$) or we do not observe whether the egg is fresh or not ($OF = unknown$).
- F: the "real" state of the egg, fresh ($F = t$) or rotten ($F = f$).
- O: whether we get a six-egg omelette ($O = 6O$), a five-egg omelette ($O = 5O$) or no omelette at all ($O = nO$).
- S: whether an egg is spoiled ($S = t$) or not ($S = f$).

Let us note that, for example, the edge between *BAC* and C represents a causal influence of the value of the decision *BAC* upon the value of the state C while the edge from *OF* to *PIO* denotes an informational influence which expresses the fact that the value of the variable *OF* will be observed before choosing the value of the variable *PIO*. Note that the example could have been represented without the nodes C, O and S but they will be useful to illustrate variable elimination procedures.

The DAG structure, together with the underlying assumptions allow to define a strict partial ordering of the variables in $\mathcal{X} \cup \mathcal{D}$, described as a partition $\Omega = \{I_0, \ldots, I_p\}$ of the state variables. Indeed, if $\{D_1, \ldots, D_p\}$ is the *ordered* set of decision variables, $I_{j-1} \subseteq \mathcal{X}$ (for $1 \leqslant j \leqslant p$) represents the set of state variables which values are revealed "just before" decision D_j is chosen. In the problem of sequential decision under uncertainty modelled as a PID, the values of the variables in I_0 are revealed first, then the decision d_1 is taken, then the values of the variables in I_1 are revealed, then the decision d_2 is taken, etc. The values of the variables in I_p are not revealed before all the decision are taken (or more exactly their revelation has no influence on decisions). Formally, the sets I_j are determined from the DAG:

$$I_0 = Par(D_1),$$
$$I_j = Par(D_{j+1}) - (I_0 \cup \cdots \cup I_{j-1}), \quad \forall j = 1 \ldots p-1 \quad \text{and}$$
$$I_p = \mathcal{X} - (I_0 \cup \cdots \cup I_{p-1}).$$

In the following, methods for solving possibilistic decision problems modelled as a PID will be based on a complete ordering of chance and decision nodes, *compatible with the DAG*. This notion of DAG-compatible ordering is defined from the partition Ω deduced from the DAG:

Definition 1 *(DAG-compatible ordering).* '$\prec$', a complete ordering of all state and decision nodes, is said to be *DAG-compatible* if and only if:

- for two decision nodes D_i and D_j, $D_i \prec D_j$ iff $i < j$,
- for two chance nodes $Y \in I_i$ and $Y' \in I_j$, $Y \prec Y'$ if $i < j$,
- for a chance node $Y \in I_i$ and a decision node D_j, $Y \prec D_j$ iff $i < j$.

Example 2. For the PID graphically represented in Fig. 1, the partitioning defined in a unique way from the DAG is $I_0 = \emptyset$, $I_1 = \{OF\}$ and $I_2 = \{C, F, S, O\}$ since the ordering between decision variables is $BAC \prec PIO$. Indeed, *BAC* has no predecessor, *OF* is the only predecessor of *PIO* and the values of the other state variables will not be revealed before every decision will be taken (except for C, whose value has no influence on the decision *PIO*).

The strict complete ordering $BAC \prec OF \prec PIO \prec F \prec O \prec C \prec S$ is an example of a DAG-compatible ordering for this PID.

3.2. Semantics of a PID

In addition to the graphical part, which is identical to that of classical influence diagrams, PID also comprise numerical specifications. In the possibilistic setting, this numerical ordering is handled in a qualitative way. A conditional possibility table $\pi_{X_i}(X_i|Par(X_i))$ is attached to each state variable X_i. The set of conditional possibility tables is $\Phi = \{\pi_{X_i}(x_i|x_{Par(X_i)}, d_{Par(X_i)})\}_{x_i \in X_i}$, where $x_{Par(X_i)} \in \mathcal{X}_{Par(X_i)} = \bigotimes_{X_j \in (Par(X_i) \cap \mathcal{X})} X_j$ and $d_{Par(X_i)} \in \mathcal{D}_{Par(X_i)} = \bigotimes_{D_j \in (Par(X_i) \cap \mathcal{D})} D_j$. If X_i is a root of the DAG ($Par(X_i) = \emptyset$) we specify the a priori possibility degrees $\pi_{X_i}(x_i) \in L$ associated with each value x_i of X_i.

Example 3. In the example of Fig. 1, let us take $L = \{0, \ldots, 5\}$. The conditional possibilities are defined w.r.t. the natural meaning of the problem. The only possibilities that can not be determined directly using the problem statement are the a priori possibilities of F. Let us take $\pi_F(F = t) = 5$ and $\pi_F(F = f) = 3$ (which describes the assumption that the egg is expected to be fresh rather than rotten). Then, we obtain the following conditional and a priori possibility tables:

$\pi_C(C\|BAC)$	y	n
t	5	0
f	0	5

$\pi_S(S\|F, PIO)$	(t, n)	(t, y)	$(f, \cdot)$
t	5	0	0
f	0	5	5

$\pi_{OF}(OF\|BAC, F)$	(y, t)	(y, f)	$(n, \cdot)$
t	5	0	0
f	0	5	0
u	0	0	5

$\pi_O(O\|F, PIO)$	$(\cdot, n)$	(t, y)	(f, y)
$6O$	0	5	0
$5O$	5	0	0
nO	0	0	5

$\pi_F(F)$	*a priori*
t	5
f	3

Once a decision $d \in \mathcal{D}$ is fixed, the chance nodes of the PID form a possibilistic causal network and determine a unique *less specific* joint possibility distribution π on the interpretations over chance nodes $x = \{x_1, \ldots, x_n\}$. This joint possibility distribution can be computed by applying the *chain rule* [1]:

$$\pi(x|d) = \min_{X_i \in \mathcal{X}_i} \pi_{X_i}(x_i|x_{Par(X_i)}, d_{Par(X_i)}), \quad \forall x \in \mathcal{X}, \forall d \in \mathcal{D}. \tag{3}$$

We also define the set $\Psi = \{\mu_k(x_{Par(V_k)}, d_{Par(V_k)})_{V_k \in \mathcal{V}}\}$. For each utility node V_k, we prescribe ordinal values $\mu_k(x_{Par(V_k)}, d_{Par(V_k)})$ to every possible instantiations $(x_{Par(V_k)}, d_{Par(V_k)})$ of the parent variables of V_k. These values represent satisfaction degrees attached to the local instantiation of the parent variables. It is assumed, analogously to *Flexible Constraint Satisfaction Problems* [5], that the global satisfaction degree $\mu(x,d)$ associated with a global instantiation (x,d) of all variables (state and decision) can be computed as the *minimum* of the local satisfaction degrees:

$$\mu(x,d) = \min_{k=1\ldots q} \mu_k(x_{Par(V_k)}, d_{Par(V_k)}), \quad \forall x \in \mathcal{X}, \forall d \in \mathcal{D}. \tag{4}$$

Example 4. In the example of Fig. 1, utility functions μ_O, μ_S and μ_R are defined as follows, and for any global instantiation (x,d), $\mu(x,d) = \min\{\mu_O(o), \mu_S(s), \mu_C(c)\}$.

O	$\mu_O(O)$
$6O$	5
$5O$	3
nO	0

S	$\mu_S(S)$
f	5
t	2

C	$\mu_C(C)$
f	5
t	4

We are now interested in computing the optimistic and pessimistic utilities of a strategy δ. A strategy δ for a PID specifies the value d of all the decision variables knowing the value x of the state variables ($d = \delta(x)$). Thus, the possibility distribution π_δ can be computed using (3):

$$\pi_\delta(x) = \pi\big(x|\delta(x)\big) = \min_{k=1\ldots n} \pi_{X_k}(x_k | x_{Par(X_k)}, d^x_{Par(X_k)}), \tag{5}$$

where $d^x = \delta(x)$ is the value of all decision variables, uniquely defined from δ and x.

In the same way, a global satisfaction degree $\mu_\delta(x)$ on state variables instantiations induced by a strategy can also be computed from (4):

$$\mu_\delta(x) = \mu\big(x, \delta(x)\big) = \min_{l=1\ldots q} \mu_{V_l}(x_{Par(V_l)}, d^x_{Par(V_l)}). \tag{6}$$

$\pi_\delta(x)$ and $\mu_\delta(x)$ thus depend on the value of state variables only. It is then possible to compute the optimistic and pessimistic possibilistic utilities of a strategy δ using Eqs. (1) and (2).

3.3. Possibilistic decision tree

As it has already been noticed, the choice of the value d_j of a decision variable D_j can only depend on the values of the variables which have been previously observed. A DAG-compatible ordering on variables '$\prec$' defines, in a non-ambiguous way, the variables whose values are known when the value of the decision D_j has to be fixed. A strategy δ can then be decomposed as a set of sub-strategies $\delta_1, \ldots, \delta_p$, where $\delta_j(\mathit{Predecessors}(D_j))$ specifies the value of the decision variable D_j, knowing the observed values of all the variables in $\mathit{Predecessors}(D_j) = \{Y \in \mathcal{X}, Y \prec D_j\}$.

A priori, nothing prevents $\mathit{Predecessors}(D_j)$ to contain decision variables. However, for a strategy $\delta = \{\delta_1, \ldots, \delta_p\}$, if $D_i \in \mathit{Predecessors}(D_j)$, it is possible to replace $\mathit{Predecessors}(D_j)$ with $(\mathit{Predecessors}(D_j) - \{D_i\})$ since the value of D_i is determined from the values of the variables of $\mathit{Predecessors}(D_i) \subseteq \mathit{Predecessors}(D_j)$. By repeating this operation until $\mathit{Predecessors}(D_j)$ does not contain anymore decision variables, δ is replaced with an equivalent strategy δ' in which no subset $\mathit{Predecessors}(D_j)$ contains any decision variable.

To solve sequential decision under uncertainty problems modelled as a PID, it is useful to show an equivalent representation in term of a *decision tree*. A PID can be unfolded into a *possibilistic decision tree*, using a method very similar to that of [11] for usual ID. A decision tree represents every possible scenarios of successive actions and observations, for a fixed *DAG-compatible ordering*. In a possibilistic decision tree, branches issuing from a decision node are labelled with the corresponding instantiation of the decision variable. Branches issuing from a chance node X_i are labelled with instantiations of the variable together with their possibility degree computed from the local transition possibilities and the values of the ancestors of X_i in the branch. Fig. 2 shows a decision tree obtained by unfolding the PID in Fig. 1 with the DAG-compatible ordering $BAC \prec OF \prec PIO \prec F \prec O \prec S \prec C$.

A decision tree is gradually built by successively considering the state and decision variables with respect to the DAG-compatible ordering. The root of the tree is the first variable, then branches issuing from a node are added and

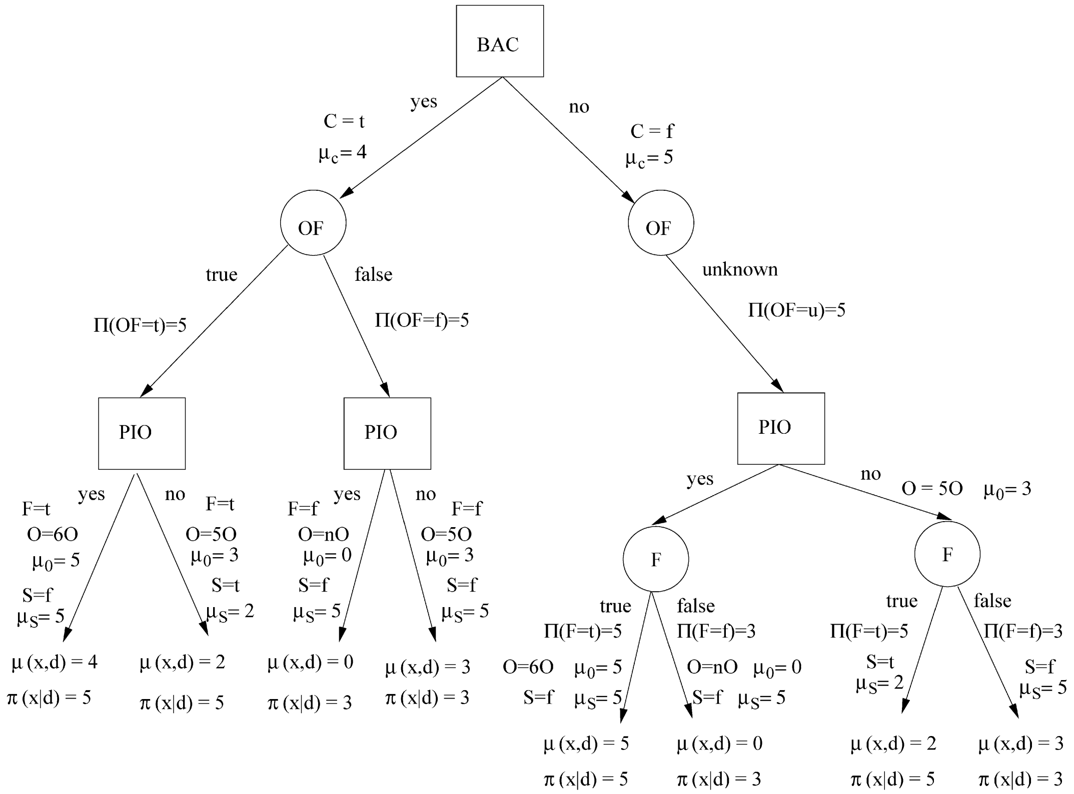

Fig. 2. Decision tree associated with the example PID.

labelled with the different possible values of the variable corresponding to the node. Once the tree is completely built, the branches of the tree correspond to complete instantiations (x, d) of the state and decision variables. Then, to each leaf are attached both a possibility degree $\pi(x|d)$ and an aggregated satisfaction degree $\mu(x, d)$. Following the chain rule (Eqs. (3) and (4)), $\pi(x|d)$ (resp. $\mu(x, d)$) is the minimum of the transition possibilities (resp. satisfaction degrees) involved in the branch from the root to the leaf.

In the decision tree of Fig. 2, after nodes *BAC*, *OF* and *PIO* have been added, node F is omitted in the left part of the tree since it has only one possible value, given the observed value of *OF* (for example, in the leftmost branch F is *true* since *OF* is *true*). For the same reason, nodes C, O and S are not displayed. Then, it can be easily checked that, to each leaf of the decision tree, is associated one particular interpretation of the variables. In Fig. 2, the leftmost branch corresponds to the instantiation $\{BAC = yes, OF = t, PIO = yes, F = t, O = 6O, S = f, C = t\}$. Its possibility degree is $\pi = \pi_{OF}(OF = t|BAC = yes, F = t) = 5$ and its utility is $\mu = \min(\mu_O(6O), \mu_C(t), \mu_S(f)) = 4$.

It has to be noticed that the number of leaves of a possibilistic decision tree is equal to the number of (authorised) instantiations (x, d) (such that $\pi(x|d) \neq 0$) of every state and decision variables. This representation is then more expensive in space, in general, than the representation using a PID. It can also be shown that two decision trees built from the same PID and two different DAG-compatible orderings will see the same pairs $\{\mu(x, d), \pi(x, d)\}$ attached to leaves corresponding to identical complete instantiations (x, d). The trees corresponding to two different DAG-compatible orderings will of course be different, though.

3.4. Strategies, possibilistic values of strategies

A possibilistic decision tree can be used to represent a strategy for a PID and for computing the optimistic and pessimistic utilities of this strategy. Namely, a strategy representation can be obtained from the decision tree by pruning every branches issuing from each decision node, but one. The remaining branch then represents the decision node value associated by the local strategy to the corresponding instantiation of the ancestor variables of the decision node.

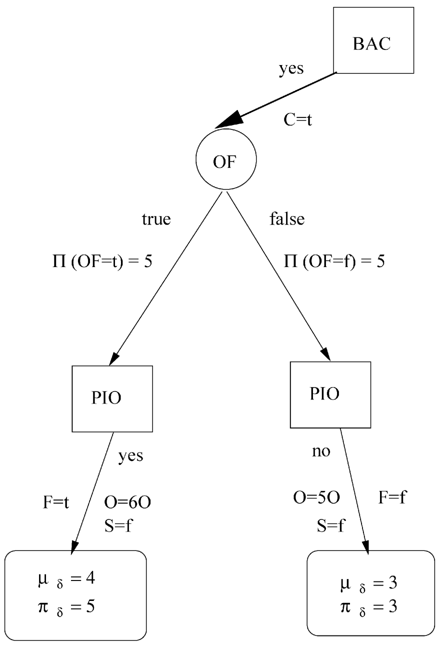

Fig. 3. Strategy representation from decision tree.

Example 5. Let us now consider the following strategy $\delta = \{\delta_{BAC}, \delta_{PIO}\}$: $\delta_{BAC} = yes$ (*BAC* has no parent, so δ_{BAC} is unconditional), $\delta_{PIO}(OF = t) = yes$ and $\delta_{PIO}(OF = f) = no$.

Fig. 3 represents the strategy $\delta = \{\delta_{BAC}, \delta_{PIO}\}$ of Example 5. The branch "no" issuing from *BAC* has been pruned, which amounts to suppressing the right part of the decision tree of Fig. 2 and, in the same way, only one branch issuing from each copy of node *PIO* has been kept.

The possibility and utility levels $\pi_\delta(x)$ and $\mu_\delta(x)$ of the leaves can then be used to compute the optimistic and pessimistic utilities of strategy δ described in Example 5:

Example 6 *(Continued).* It can be easily checked that only the two instantiations $x_1 = (F = t, O = 6O, S = f, OF = t, C = t)$ and $x_2 = (F = f, O = 5O, S = f, OF = f, C = t)$ have a non-zero possibility degree: $\pi_\delta(x_1) = \pi_F(F = t) = 5$ and $\pi_\delta(x_2) = \pi_F(F = f) = 3$. And we can check that $\mu_\delta(x_1) = 4$ and $\mu_\delta(x_2) = 3$. So,

$$u^*(\delta) = \max\bigl\{\min\bigl(\pi_\delta(x_1), \mu_\delta(x_1)\bigr), \min\bigl(\pi_\delta(x_2), \mu_\delta(x_2)\bigr)\bigr\} = 4 \quad \text{and}$$
$$u_*(\delta) = \min\bigl\{\max\bigl(n\bigl(\pi_\delta(x_1)\bigr), \mu_\delta(x_1)\bigr), \max\bigl(n\bigl(\pi_\delta(x_2)\bigr), \mu_\delta(x_2)\bigr)\bigr\} = 3.$$

3.5. Strategy optimisation

Solving a PID amounts to finding an *optimal* strategy δ^*, i.e. one which maximises the (either optimistic or pessimistic) possibilistic utility. We present next a tree search method, based on the decision tree built from the PID, that can be used to compute such optimal strategies. Describing the method will be useful in order to introduce the complexity results which will be shown in Section 4. In Section 5, we will present algorithms based on a variable elimination procedure that can be used in practice to solve a PID in the optimistic or pessimistic framework.

In order to compute a possibilistic optimal strategy from the decision tree, we can apply a *depth-first search* method, allowing to attach a utility degree to each node, and to prune branches issuing from each decision node by keeping only one branch. The two basic principles of the computation are the following:

- If all the children of a decision node already have an attached possibilistic utility, only the issuing branch with the highest value is kept. This highest value is attached to the decision node.

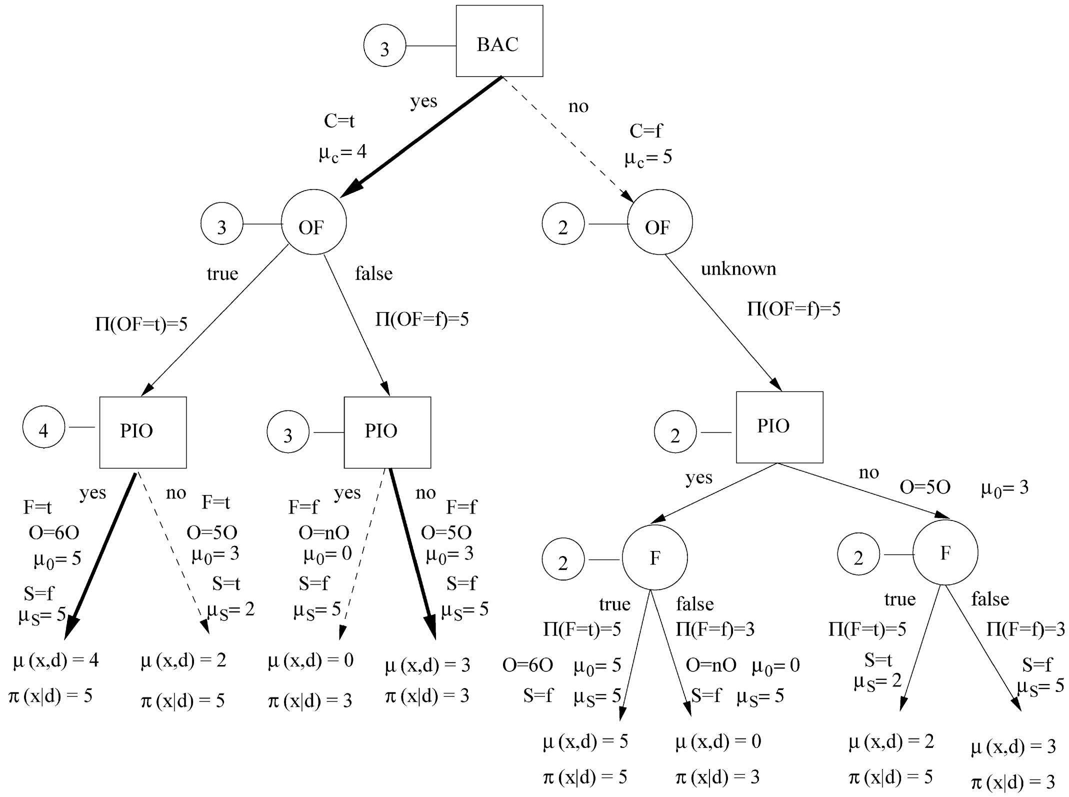

Fig. 4. Optimal pessimistic decision strategy computation.

- If all the children of a chance node have an attached possibilistic utility, its possibilistic (either optimistic or pessimistic) utility is computed.

The depth-first search is made until a possibilistic utility is attached to the root node. The remaining tree represents an optimal strategy, together with its utility (optimistic or pessimistic according to the chosen case). Fig. 4 shows how an optimal pessimistic strategy can be computed for the omelette example. The circled numbers attached to decision and chance nodes represent the possibilistic pessimistic utilities which have been successively computed during the search. Note that the optimal pessimistic strategy, which is the one described in Fig. 3, consists in breaking the egg apart in a cup for inspection, and to put it in the omelette if it is observed to be fresh, and to throw it away if it is not.

3.6. Remarks on optimal strategies

The above-described computation method must explore a decision tree of size exponential in the number of variables of the PID in order to compute an optimal strategy. It may not be necessary to keep it all in memory in order to compute the utility of the best strategy. Indeed, in the next section we will show that computing the utility of an optimal (pessimistic or optimistic) strategy requires polynomial space. However, optimal strategies may require space exponential in the number of decision variables to be expressed: they can be quite large subtrees of the decision tree.

In the optimistic case, however, it can be shown that one can always find *unconditional* optimal strategies (sequences of decisions). Indeed, in order to compute an optimal strategy, it is enough to find in the decision tree the branch (x, d) which maximises $\min(\pi(x|d), \mu(x, d))$. The corresponding instantiation of decision variables d is an unconditional optimistic optimal strategy.

Example 7. In Fig. 4, the branch $\{(BAC = no), (C = false), (OF = unknown), (PIO = yes), (F = true), (O = 6O), (S = false)\}$ has a maximal utility of 5. It can be checked that the unconditional strategy δ^*, $\delta^*_{BAC} = no$, $\delta^*_{PIO} = yes$

has utility $u^*(\delta^*) = 5$. Let us note that this strategy is better, according to the optimistic criterion, than the one of Example 5 which has utility $u^*(\delta) = 4$.

However, such unconditional strategies lack a desirable property shared by all optimal strategies in stochastic ID (and by pessimistic optimal strategies in PID): they are not *dynamically consistent*. Dynamic consistency of an optimal strategy means that the strategy remains optimal in time, when state variables are successively instantiated [21]. It may happen that an unconditional optimal optimistic strategy does not remain optimal if the course of nature is different from what was anticipated. So, even if they can require exponentially less space to be expressed than conditional strategies, the unconditional optimal optimistic strategies may have to be recomputed whenever a chance variable value is observed which differs from the one in the branch that allowed to compute the current strategy. This would give a motivation for designing a real-time algorithm in the optimistic case but we do not focus on this point in this article.

In the next section we discuss more formally, in terms of computational complexity considerations, this difference in the difficulty of solving optimistic or pessimistic optimisation problems. Then, in Section 5 we will describe a generic variable elimination algorithm which can be used to compute optimal dynamically consistent strategies for both types of optimisation problems.

4. Complexity results

In this section, we formalise the intuition which results from the algorithm that has been exposed in the previous section, namely the intuition that optimistic optimisation problems are "easier" to solve than pessimistic ones.

We will consider here the two decision problems:

Definition 2 *(Optimistic strategy optimisation (OSO)).*
INSTANCE: A PID $\mathcal{P} = (\mathcal{X}, \mathcal{D}, \mathcal{V}, \Phi, \Psi)$, a level $\alpha \in L$
QUESTION: Does there exist δ, $u^*(\delta) \geqslant \alpha$, where u^* is the optimistic utility function associated to the PID $\mathcal{P}$?

Definition 3 *(Pessimistic strategy optimisation (PSO)).*
INSTANCE: A PID $\mathcal{P} = (\mathcal{X}, \mathcal{D}, \mathcal{V}, \Phi, \Psi)$, a level $\alpha \in L$
QUESTION: Does there exist δ, $u_*(\delta) \geqslant \alpha$, where u_* is the pessimistic utility function associated to the PID $\mathcal{P}$?

4.1. Complexity of the optimistic strategy optimisation problem

Proposition 1 *(Optimistic strategy optimisation (OSO)). A PID $\mathcal{P}$ and a level $\alpha \in L$ being given, the problem of deciding whether there exists a strategy δ that attains utility degree α ($u^*(\delta) \geqslant \alpha$) is NP-complete.*

Proof. 1) OSO is in NP:

OSO writes: $\big(\exists \delta, \max_x \min\big(\pi_\delta(x), \mu_\delta(x)\big) \geqslant \alpha\big)$?

But $\big(\exists \delta, \max_x \min\big(\pi_\delta(x), \mu_\delta(x)\big) \geqslant \alpha\big) \Leftrightarrow \big(\exists d \exists x \min\big(\pi(x|d), \mu(x, d)\big) \geqslant \alpha\big)$,

where $\pi(x|d) = \min_{i=1\ldots n} \Pi_{X_i}(x_i|x_{Par(X_i)}, d_{Par(X_i)})$ and $\mu(x,d) = \min_{l=1\ldots q} \mu_{V_l}(x_{Par(V_l)}, d_{Par(V_l)})$.

This question can be answered through an oracle guess:
Guess (x, d) *such that*:

$$\min\Big(\min_{i=1\ldots n} \Pi_{X_i}(x_i|x_{Par(X_i)}, d_{Par(X_i)}), \min_{l=1\ldots q} \mu_{V_l}(x_{Par(V_l)}, d_{Par(V_l)})\Big) \geqslant \alpha.$$

Such a guessed solution can be checked in time polynomial in the size needed to express the possibility and utility tables, and it is easy to see that a solution corresponds to an unconditional strategy δ^* ($\delta_i^*(x) = d_i$, $\forall x$).

2) NP-hardness: In order to show that OSO is NP-hard, it is enough to exhibit a polynomial-time reduction from 3-SAT to OSO (since 3-SAT is NP-complete). Let a 3-SAT instance be defined as a CNF formula $F(x) = F_1(x) \wedge \cdots \wedge F_q(x)$, where the F_k are 3-clauses. This instance can be represented as a $q \times 3$ matrix of integers, element

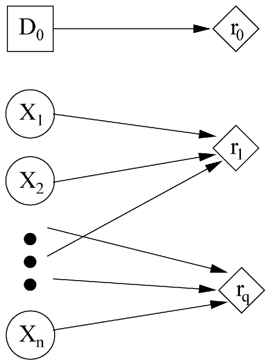

Fig. 5. PID corresponding to a polynomial transformation of 3-SAT.

$(k, j) \in \{1 \dots q\} \times \{1, 2, 3\}$ of the matrix being the index of the jth variable of the kth clause, negated if the variable is negated in the clause. It thus requires space $q \log(n)$ to represent $F(x)$.

Let us define the following PID $\mathcal{P} = (\mathcal{X}, \mathcal{D}, \mathcal{V}, \Phi, \Psi)$ described in Fig. 5:

- $\mathcal{X} = \{X_1, \dots, X_n\}$ are the binary variables involved in the CNF F,
- $\mathcal{D} = \{D_0\}$, where D_0 has domain $\{yes, no\}$,
- $\mathcal{V} = \{V_0, \dots, V_q\}$ contains one symbol for each clause and the additional symbol V_0.
- Φ is a set of unconditional possibility measures $\{\Pi_{X_i}\}_{X_i \in \mathcal{X}}$, one for each state variable, and $\Pi_{X_i}(x_i) = 1_L, \ \forall x_i$.
- Ψ is a set of utility functions μ_{V_k}, one for each clause F_k, with scope $Par(V_k) = Scope(F_k)$. $\mu_{V_k}(x_{Par(V_k)}) = 1_L$ if $x_{Par(V_k)}$ satisfies F_k and 0_L otherwise. Ψ contains an additional utility function, μ_0, which scope is $\{D_0\}$, and for which $\mu_0(d_0) = 1_L$ if $d_0 = yes$ and 0_L otherwise.

$$\text{Then, } \Big(\exists \delta, \max_x \min\big(\pi_\delta(x), \mu_\delta(x)\big) \geqslant 1_L\Big) \Leftrightarrow F \text{ is satisfiable.}$$

So, this OSO problems solves 3-SAT if we fix $\alpha = 1_L$. The corresponding *unconditional* strategy is $\delta : d_0 = yes$.

The transformation is obviously polynomial in time since utility tables have at most 2^3 elements (the F_k are 3-clauses) and so, OSO is NP-hard.

Papadimitriou [14] rather used *logarithmic space* transformations to show NP-completeness of problems. We can also show that the proposed transformation is logarithmic in space. We only sketch the proof, since it is not directly necessary to our proof of NP-hardness. Note first that we only need to build the functions $\{\mu_{V_k}\}_{k=1\dots q}$ and their scopes $\{Par(V_k)\}_{k=1\dots q}$. The transition possibilities $\{\pi_{X_i}\}_{i=1\dots n}$ can remain implicit (we know that $\pi_{X_i}(x_i) = 1_L, \ \forall i, \forall x_i$ and writing them in a *write-only* tape would require a counter of size $\log(n)$). The scopes $\{Par(V_k)\}_{k=1\dots q}$ can be stored as lines of a $q \times 3$ matrix which is simply the absolute value of $F(x)$ and which can be written using only "constant" space, by copying matrix $F(x)$ bit by bit, forgetting the sign bits. In order to write the μ_{V_k} functions, we can notice that $\mu_{V_k}(x_{Par(V_k)}) = 0_L$ if $x_{Par(V_k)}$ is exactly the opposite of the signs of line k of $F(x)$ (literals in the clause are positive if the corresponding numbers in the corresponding line of F are negative, and vice-versa) and is 1_L otherwise. Thus, the μ_{V_k} can be written in a clever way by storing only the index of their zero element, which can again be done in constant time by reading the signs of the literals in F_k and writing the corresponding bits of the index accordingly (0 if sign is negative, 1 otherwise). The full transformation is of course *log-space* (it is even *constant-space*). $\square$

4.2. *Complexity of the pessimistic strategy optimisation problem*

The pessimistic strategy optimisation problem is more complex since the corresponding exploration of the (complete) decision tree alternates maximisations at decision nodes and minimisations at chance nodes. The following result can be shown:

Proposition 2 *(Pessimistic strategy optimisation (PSO)). A PID $\mathcal{P}$ and a utility degree α being given, the problem of deciding whether there exists a strategy δ which attains utility degree α ($u_*(\delta) \geqslant \alpha$) is PSPACE-complete.*

Proof. 1) PSO is in PSPACE:

PSO writes: $\big(\exists\delta, \min_x \max\big(n\big(\pi_\delta(x)\big), \mu_\delta(x)\big) \geqslant \alpha\big)$?

This also writes: $\big(\max_\delta \min_x \max\big(n\big(\pi_\delta(x)\big), \mu_\delta(x)\big) \geqslant \alpha\big)$?

Given a state variables partition $\Omega = \{I_0, \ldots, I_p\}$ defined from an ordering of variables '$\prec$', DAG-compatible for the PID $\mathcal{P}$,[2] the PSO can also be written (local strategy δ_j only depends on the variables instantiated before d_j):

$$\min_{x_{I_0}} \max_{d_1} \min_{x_{I_1}} \ldots \max_{d_p} \min_{x_{I_p}} \max\big(n\big(\pi(x|d)\big), \mu(x,d)\big) \geqslant \alpha,$$

Finally, this can be transformed into:

$$\forall x_{I_0} \exists d_1 \forall x_{I_1} \ldots \exists d_p \forall x_{I_p} \big(\max\big(n\big(\pi(x|d)\big), \mu(x,d)\big) \geqslant \alpha\big).$$

This can be seen as a two-players game, where Player I chooses the value of decision variables and Player II chooses the values of chance variables. The game is made *alternating* by adding quantified dummy variables ($\exists z_k$) or ($\forall w_k$) not appearing in the max expression, where needed, so that the "quantification" part of the expression alternates existentially and universally quantified variables. Each player then alternates *moves* (choices for chance or decision variables) and when all variables have been instantiated, the final *position* is labelled *WIN* for Player I if the expression in brackets is true, and *WIN* for Player II if not.

This game is PSPACE [14] since:

- The length of any legal sequence of moves is bounded, by $2 \times (n + p)$ in the worst case (when only I_0 or only I_P is non-empty).
- Any *final* position (i.e. when all variables have been instantiated) can be labelled *WIN* for Player I or II in polynomial space. In fact, the expression in brackets can even be evaluated in polynomial time.
- Finally, given any arbitrary position of the game, given by the instantiation of the k first variables, the set of possible moves for the current player to play and the set of possible following positions can be constructed using polynomial space only. This fact is obvious, since the next possible plays are determined by checking the (polynomial size) set of allowed values for the next variable to instantiate.

The conclusion is that PSO, which can be rewritten as the above described two-players game, is also in PSPACE.

2) PSPACE-hardness: Here we show how a *Quantified Boolean Formula* (3-QBF, in fact) problem can be reduced to a PSO problem. Let us consider the 3-QBF $\exists y_1 \forall y_2 \exists y_3 \ldots Q_n y_n F(y)$ where F is a CNF formula ($F(y) = F_1(y) \wedge \cdots \wedge F_q(y)$, where the F_i are 3-clauses). Let us now define a PID $\mathcal{P} = (\mathcal{X}, \mathcal{D}, \mathcal{V}, \Phi, \Psi)$ from the 3-QBF in the following way:

- $\mathcal{X}$ is the set of universally quantified variables in the 3-QBF. Variables X_i are all binary.
- $\mathcal{D}$ is the set of existentially quantified variables in the 3-QBF. Variables D_j are all binary.
- $\mathcal{V}$ contains one symbol V_k for each clause in the 3-QBF.
- Φ is a set of unconditional possibility measures Π_{X_i}, one for each state variable, and $\Pi_{X_i}(x_i) = 1_L,\ \forall x_i$.
- Ψ is a set of utility functions, one for each clause F_k, and $\mu_{V_k}(x_{Par(V_k)}, d_{Par(V_k)}) = 1_L$ if the corresponding instantiation $y_{Scope(F_k)}$ satisfies F_k and 0_L else.

Fig. 6 shows the graphical part of the PID corresponding to the example 3-QBF: $\exists y_1 \forall y_2 \exists y_3 \forall y_4 \exists y_5 ((y_1 \vee \neg y_3 \vee y_5) \wedge (\neg y_2 \vee y_3 \vee y_4) \wedge (\neg y_4 \vee \neg y_5))$.

Now, we can show that in general a 3-QBF $\exists y_1 \forall y_2 \exists y_3 \ldots Q_n y_n F(y)$ is true if and only if the corresponding PSO is true:

$$\big(\max_\delta u_*(\delta) \geqslant 1_L\big)?$$

In order to prove this, just look at the following equivalences:

[2] Let us recall that Ω is the same for every DAG-compatible ordering '$\prec$'.

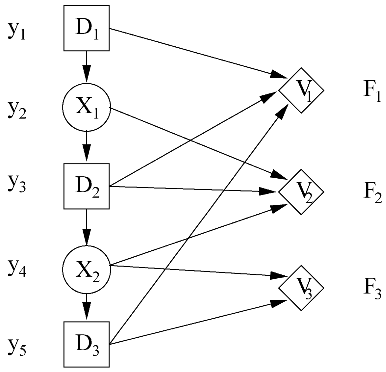

Fig. 6. PID corresponding to the QBF $\exists y_1 \forall y_2 \exists y_3 \forall y_4 \exists y_5 ((y_1 \vee \neg y_3 \vee y_5) \wedge (\neg y_2 \vee y_3 \vee y_4) \wedge (\neg y_4 \vee \neg y_5))$.

$$\max_\delta u_*(\delta) \geqslant 1_L \Leftrightarrow \max_{d_1} \min_{x_1} \max_{d_2} \ldots \max\bigl(n\bigl(\pi(x|d)\bigr), \mu(x,d)\bigr) \geqslant 1_L,$$
$$\text{where } \pi(x|d) = \min_{X_i} \Pi_{X_i}(x_i) = 1_L, \quad \forall x$$
$$\text{and } \mu(x,d) = \min_{k=1\ldots q} \mu_k(x_{Par(V_k)}, d_{Par(V_k)}), \quad \forall x.$$
$$\text{So, } \max_\delta u_*(\delta) \geqslant 1_L \Leftrightarrow \max_{d_1} \min_{x_1} \max_{d_2} \ldots \mu(x,d) \geqslant 1_L, \text{ and}$$
$$\max_\delta u_*(\delta) \geqslant 1_L \Leftrightarrow \exists d_1 \forall x_1 \exists d_2 \ldots \min_{k=1\ldots q} \mu_k(x_{Par(V_k)}, d_{Par(V_k)}) \geqslant 1_L.$$

Noticing that $\min_{k=1\ldots q} \mu_k(x_{Par(V_k)}, d_{Par(V_k)}) \geqslant 1_L \Leftrightarrow F(x,d)$ is true, and rewriting the x_i and d_j as y_k variables, we get

$$\max_\delta u_*(\delta) \geqslant 1_L \Leftrightarrow \exists y_1 \forall y_2 \exists y_3 \ldots F(y).$$

So, this PSO problem "solves" 3-QBF, and thus 3-QBF has been reduced to PSO. Furthermore, the transformation is polynomial in time and space since utility tables have at most 2^3 elements (the F_k are 3-clauses). The proposition is then proved: PSO is PSPACE-complete. □

4.3. Complexity of unobservable pessimistic strategy optimisation (UPSO)

In [10], it was wrongly stated that PSO is Σ_2^p-complete. However, this result holds in a particular case. Indeed, in the *unobservable case*, where decisions are to be made *before* any chance variable value is observed, the complexity of the corresponding *unobservable pessimistic strategy optimisation* (*UPSO*) problem falls down.[3] The next proposition can then be shown.

Proposition 3 *(Unobservable pessimistic strategy optimisation (UPSO)). A PID being given, the problem of computing an optimal pessimistic strategy δ^* and its utility $u_*(\delta)$, when state variables are never observed is Σ_2^p-complete.*

Sketch of proof. In the unobservable case, it can be shown that since state variables are only observed after all decisions have been chosen, Eq. (2) writes:

$$\max_{d_1} \ldots \max_{d_p} \min_x \max\bigl(n\bigl(\pi(x|d)\bigr), \mu(x,d)\bigr) \geqslant \alpha,$$

which can be rewritten:

$$\exists d_1, \ldots, d_p, \forall x_1, \ldots, x_n \bigl(\max\bigl(n\bigl(\pi(x|d)\bigr), \mu(x,d)\bigr) \geqslant \alpha\bigr).$$

[3] The UPSO case is the one of a PSO problem on a PID for which $I_0 = I_1 = \cdots = I_{p-1} = \emptyset$ and $I_p = \mathcal{X}$.

This question is of the form "is it true that $\exists a, \forall b, r(a, b)$?", which is representative of the class of Σ_2^p-complete problems in the polynomial hierarchy. In the same way as OSO was shown to be NP-complete and PSO PSPACE-complete, UPSO can be shown to be Σ_2^p-complete. □

5. Variable elimination algorithms

In this section we present *variable elimination* algorithms for computing optimal strategies for PID with either optimistic or pessimistic utilities. These algorithms are inspired by the eponymous algorithm for stochastic ID, initially proposed by [23] and also used for example in [12]. The basic principle of the algorithms is to remove one by one the nodes of the influence diagram in order to obtain an optimal strategy together with its attached utility. The variables will have to be eliminated following a *DAG-compatible* ordering for the PID.

[15] defines four "licit" operations on the nodes of the DAG attached to an influence diagram. These operations allow to simplify the structure of the influence diagram without modifying the optimal strategy or its value. These operations are the following:

(1) "Barren node removal". It is possible to remove the state or decision nodes which have no successors since the values of the variables corresponding to the other nodes are not affected by this removal.
(2) "Conditional expectation" (chance node removal). It is possible to remove a chance node directly preceding (only) utility nodes. This removal implies a modification of the corresponding utility function: the "parent" variables of the removed chance node are incorporated in the scope of the utility function.
(3) "Maximisation" (decision node removal). It is possible to remove a decision node directly preceding (only) utility nodes and to compute a new equivalent PID together with a local strategy associated to the removed node. The "parent" variables of the removed decision node are incorporated in the scope of the utility function.
(4) "Arc reversal". The removal of a chance node, needed due to the variable elimination ordering at a given time step, can be illicit when this node is parent of another chance node. To solve this issue, a licit operation of arc reversal is defined.

In this section, we define four operations of this type in the PID framework. As in the framework of influence diagrams, these operations will be applied when necessary, by an algorithm in which all the variables are successively removed following a DAG-compatible ordering.

In our framework, it is obvious that the values of the strategies are not affected by the first transformation (barren node removal). This transformation is applied as soon as the current node to remove will be without any successor. The second and third transformations are applied as soon as the current node to eliminate contains only utility nodes as successors. When the node to eliminate is a decision node, a corresponding local strategy is computed. The last transformation (arc reversal) is only applied when the current node to eliminate is a chance node which contains at least another chance node as successor.[4] The computed arc reversal(s) allow then the current node to be parent of only utility nodes. It can then be eliminated by applying the second transformation.

We will show in the following that each of the four mentioned operations, when it is applied in a legal way to the current variable to eliminate, transforms the PID $\mathcal{P}$ into an *equivalent* PID $\mathcal{P}'$ (that is, with identical optimal strategy and maximal utility). Moreover, each decision variable elimination D_j allows the computation of a corresponding optimal sub-strategy $\delta^*_{D_j}$.

5.1. Barren node removal

It is simple to show that if a state variable has no successor in a PID, it can be modified without modifying the value of any strategy. Therefore, state variables without successors will be removed, this leading to remove the corresponding conditional possibility table.

Let us show this in the optimistic case (the proof is similar in the pessimistic case). Let $\mathcal{P}$ be a PID and X_i a state variable without any successor in the associated DAG. Let δ be an arbitrary strategy. Let us recall first that δ can be

[4] If the nodes are removed following a DAG-compatible ordering, when the current node to eliminate is a decision node, it cannot contain state or decision nodes as successors.

decomposed into a set $\{\delta_j\}$ of strategies deciding the value of the decision variables D_j knowing the values of their predecessors. Since X_i has no successor, none of the local strategies δ_j depends on the values of X_i.

Now, let us write the optimistic utility $u^*(\delta)$:

$$u^*(\delta) = \max_{x \in \mathcal{X}} \min\Bigl(\min_{k=1\ldots n} \pi_{X_k}(x_k|x_{Par(X_k)}, d^x_{Par(X_k)}), \min_{l=1\ldots q} \mu_{V_l}(x_{Par(V_l)}, d^x_{Par(V_l)})\Bigr).$$

This can also be written[5]

$$\begin{aligned} u^*(\delta) &= \max_{x \in \mathcal{X}} \min\Bigl(\pi_{X_i}(x_i|\ldots), \min_{\substack{k=1\ldots n \\ k \neq i}} \pi_{X_k}, \min_{l=1\ldots q} \mu_{V_l}\Bigr), \\ &= \max_{x_1,\ldots,x_{i-1},x_{i+1},\ldots,x_n} \min\Bigl(\bigl(\max_{x_i} \pi_{X_i}(x_i|\ldots)\bigr), \min_{k \neq i} \pi_{X_k}, \min_{l=1\ldots q} \mu_{V_l}\Bigr). \end{aligned}$$

But since, by definition, $\pi_{X_i}(x_i|\ldots)$ is normalised, $(\max_{x_i} \pi_{X_i}(x_i|\ldots)) = 1_L$ and

$$u^*(\delta) = \max_{x_1,\ldots,x_{i-1},x_{i+1},\ldots,x_n} \min(\min_{k \neq i} \pi_{X_k}, \min_{l=1\ldots q} \mu_{V_l}).$$

So the utility $u^*(\delta)$ does not depend on the variable X_i.

The same result can be obtained in the pessimistic case, and also in the case where the variable to eliminate is a decision variable. In the latter case, it is enough to notice that if D_j has no successor, it is not an argument of any function $\{\pi_{X_k}\}$ or $\{\mu_{V_l}\}$ and it does not participate to the computation of the utility of a strategy δ.

5.2. *Elimination of a state variable*

Recall first that any strategy δ decides on the value of each decision variable, according to its predecessor state and decision variables only. Then, if $\mathcal{X}_{end}$ denotes the set of state variables which are not the predecessors of any decision variable, the variables in $\mathcal{X}_{end}$ should be the first to eliminate in our variable elimination algorithm, since their elimination will not affect the form of the local strategies.

If $\mathcal{X}_{end}$ is non-empty, let $X_k \in \mathcal{X}_{end}$ be the next variable we wish to eliminate. Let us define:

$$\begin{aligned} &\Phi_{X_k} = \bigl\{\pi_Y\bigl(Y|Par(Y)\bigr) \in \Phi,\ X_k \in \bigl(Par(Y) \cup \{Y\}\bigr)\bigr\}, \\ &\Psi_{X_k} = \bigl\{\mu_V\bigl(Par(V)\bigr) \in \Psi,\ X_k \in Par(V)\bigr\}, \quad \text{and} \\ &\overline{\Phi_{X_k}} = \Phi - \Phi_{X_k} \quad \text{and} \quad \overline{\Psi_{X_k}} = \Psi - \Psi_{X_k}. \end{aligned} \tag{7}$$

Let us also define:

$$\phi^*_k(x, d) = \min_{\pi_{X_i} \in \overline{\Phi_{X_k}}} \pi_{X_i}(x_i|x_{Par(X_i)}, d_{Par(X_i)}), \quad \forall x, d. \tag{8}$$

Since $X_k \notin Par(X_i) \cup \{X_i\}$ for any $\pi_{X_i} \in \overline{\Phi_{X_k}}$ by definition of $\overline{\Phi_{X_k}}$, ϕ^*_k does not depend on x_k and we can write that $\phi^*_k(x, d) = \phi^*_k(x_{\bar{k}}, d),\ \forall x, d$, where $x_{\bar{k}} \in \bigotimes_{i \neq k} X_i$.

In the same way, we can also write that

$$\psi^*_k(x, d) = \psi^*_k(x_{\bar{k}}, d) = \min_{\mu_V \in \overline{\Psi_{X_k}}} \mu_V(x_{Par(V)}, d_{Par(V)}), \quad \forall x, d. \tag{9}$$

Let us note that ϕ^*_k and ψ^*_k may in general not depend on all state and decision variables apart from X_k, but only respectively on those:

- variables X_i such that $\pi_{X_i} \in \overline{\Phi_{X_k}}$ and their parents,
- parent variables of utility nodes V such that $\mu_V \in \overline{\Psi_{X_k}}$.

[5] To simplify the notations, the arguments of the functions will be omitted.

5.2.1. Optimistic case

When considering state variable elimination in the optimistic case, we will need to define also the following:

$$\psi^{-X_k}(x_{\bar{k}}, d) = \max_{x_k \in X_k} \min\Bigl(\min_{\pi_{X_i} \in \Phi_{X_k}} \pi_{X_i}(x_i | x_{Par(X_i)}, d_{Par(X_i)}), \min_{\mu_{V_j} \in \Psi_{X_k}} \mu_{V_j}(x_{Par(V_j)}, d_{Par(V_j)})\Bigr), \tag{10}$$

where ψ^{-X_k} is obviously not a function of x_k, since this variable has been marginalised out by the $\max_{x_k \in X_k}$ operator. As before, ψ^{-X_k} may depend only in general on a subset of state and decision variables, excluding X_k.

Now, we can show the next proposition:

Proposition 4. *$\forall \delta = \{\delta_j\}_{j=1\ldots p}$, where $\delta_j : \mathcal{X}_{Predecessors(D_j)} \to D_j$, $\forall X_k \in \mathcal{X}_{end}$,*

$$u^*(\delta) = \max_{x_{\bar{k}}} \min\bigl(\phi_k^*\bigl(x_{\bar{k}}, \delta(x_{\bar{k}})\bigr), \psi_k^*\bigl(x_{\bar{k}}, \delta(x_{\bar{k}})\bigr), \psi^{-X_k}\bigl(x_{\bar{k}}, \delta(x_{\bar{k}})\bigr)\bigr).$$

Proof. Indeed, if d^x denotes $\delta(x_{\bar{k}})$,[6] we have:

$$u^*(\delta) = \max_{x_1,\ldots,x_n} \min\bigl(\pi_\delta(x_1,\ldots,x_n), \mu_\delta(x_1,\ldots,x_n)\bigr)$$

$$u^*(\delta) = \max_{x_1,\ldots,x_n} \min\bigl(\min_{\pi_{X_i} \in \Phi} \pi_{X_i}(x_i | x_{Par(X_i)}, d^x_{Par(X_i)}), \min_{\mu_{V_j} \in \Psi} \mu_{V_j}(x_{Par(V_j)}, d^x_{Par(V_j)})\bigr)$$

$$\begin{aligned} u^*(\delta) = \max_{x_1,\ldots,x_n} \min\bigl(&\min_{\pi_{X_i} \in \overline{\Phi_{X_k}}} \pi_{X_i}(x_i | x_{Par(X_i)}, d^x_{Par(X_i)}), \min_{\mu_{V_j} \in \overline{\Psi_{X_k}}} \mu_{V_j}(x_{Par(V_j)}, d^x_{Par(V_j)}), \\ &\min_{\pi_{X_i} \in \Phi_{X_k}} \pi_{X_i}(x_i | x_{Par(X_i)}, d^x_{Par(X_i)}), \min_{\mu_{V_j} \in \Psi_{X_k}} \mu_{V_j}(x_{Par(V_j)}, d^x_{Par(V_j)})\bigr) \end{aligned}$$

$$\begin{aligned} u^*(\delta) = \max_{x_1,\ldots,x_{k-1},x_{k+1},\ldots,x_n} &\min\bigl(\phi_k^*\bigl(x_{\bar{k}}, \delta(x_{\bar{k}})\bigr), \psi_k^*\bigl(x_{\bar{k}}, \delta(x_{\bar{k}})\bigr), \\ &\max_{x_k} \min\bigl(\min_{\pi_{X_i} \in \Phi_{X_k}} \pi_{X_i}\bigl(x_i | x_{Par(X_i)}, \delta(x_{\bar{k}})\bigr), \min_{\mu_{V_j} \in \Psi_{X_k}} \mu_{V_j}\bigl(x_{Par(V_j)}, \delta(x_{\bar{k}})\bigr)\bigr)\bigr) \end{aligned}$$

$$u^*(\delta) = \max_{x_{\bar{k}}} \min\bigl(\phi_k^*\bigl(x_{\bar{k}}, \delta(x_{\bar{k}})\bigr), \psi_k^*\bigl(x_{\bar{k}}, \delta(x_{\bar{k}})\bigr), \psi^{-X_k}\bigl(x_{\bar{k}}, \delta(x_{\bar{k}})\bigr)\bigr). \qquad \square$$

Let us now define, for any PID $\mathcal{P} = (\mathcal{X}, \mathcal{D}, \mathcal{V}, \Phi, \Psi)$ and any variable $X_k \in \mathcal{X}_{end}$, a new PID $\mathcal{P}^{-X_k} = (\mathcal{X}', \mathcal{D}', \mathcal{V}', \Phi', \Psi')$ obtained through elimination of variable X_k:

- $\mathcal{X}' = \mathcal{X} - \{X_k\}$, $\mathcal{D}' = \mathcal{D}$ and $\mathcal{V}' = \mathcal{V} \cup \{V_{X_k}\} - \{V_j | \mu_{V_j} \in \Psi_{X_k}\}$,[7]
- $\Phi' = \overline{\Phi_{X_k}}$ and $\Psi' = \overline{\Psi_{X_k}} \cup \{\psi^{-X_k}\}$.

We can show that the optimistic utility of any strategy δ is unchanged in a PID when a chance variable belonging to the maximal class is eliminated.

Proposition 5 *(Consistency of state variable elimination). Let $\mathcal{P} = (\mathcal{X}, \mathcal{D}, \mathcal{V}, \Phi, \Psi)$ be a PID and $X_k \in \mathcal{X}_{end}$ (X_k is not an ancestor of any decision variable) and X_k is not parent of any state variable. The PID $\mathcal{P}^{-X_k} = (\mathcal{X}', \mathcal{D}', \mathcal{V}', \Phi', \Psi')$* is equivalent *to $\mathcal{P}$. In other terms, any strategy δ has the same possibilistic optimistic utility in $\mathcal{P}$ and in $\mathcal{P}^{-X_k}$.*

Proof. Let $u^*_{\mathcal{P}}(\delta)$ and $u^*_{\mathcal{P}^{-X_k}}(\delta)$ denote respectively the utility of δ in $\mathcal{P}$ and $\mathcal{P}^{-X_k}$. From Proposition 4, we have:

$$u^*_{\mathcal{P}}(\delta) = \max_{x_{\bar{k}}} \min\bigl(\phi_k^*\bigl(x_{\bar{k}}, \delta(x_{\bar{k}})\bigr), \psi_k^*\bigl(x_{\bar{k}}, \delta(x_{\bar{k}})\bigr), \psi^{-X_k}\bigl(x_{\bar{k}}, \delta(x_{\bar{k}})\bigr)\bigr).$$

---

[6] Since X_k has no descendent in the DAG except for utility nodes, it cannot be the predecessor of any decision variable and thus, since $\forall D_j$, $X_k \notin Predecessors(D_j)$, δ_{D_j} is independent of X_k.

[7] V_{X_k} represents an additional utility node in the DAG to which function ψ^{-X_k} is attached. Its parents in the DAG are the variables belonging to the scope of ψ^{-X_k}.

Now, from the definition of $\mathcal{P}^{-X_k}$, we have:

$$u^*_{\mathcal{P}^{-X_k}}(\delta) = \max_{x_1,\dots,x_{k-1},x_{k+1},\dots,x_n} \min\Big(\min_{\pi_{X_i} \in \Phi'} \pi_{X_i}(x_i | x_{Par(X_i)}, d^x_{Par(X_i)}), \min_{\mu_{V_j} \in \Psi'} \mu_{V_j}(x_{Par(V_j)}, d^x_{Par(V_j)})\Big)$$

$$u^*_{\mathcal{P}^{-X_k}}(\delta) = \max_{x_1,\dots,x_{k-1},x_{k+1},\dots,x_n} \min\Big(\min_{\pi_{X_i} \in \overline{\Phi_{X_k}}} \pi_{X_i}(x_i | x_{Par(X_i)}, d^x_{Par(X_i)}),$$
$$\min_{\mu_{V_j} \in \overline{\Psi_{X_k}}} \mu_{V_j}(x_{Par(V_j)}, d^x_{Par(V_j)}), \psi^{-X_k}\big(x_{\bar{k}}, \delta(x_{\bar{k}})\big)\Big)$$

$$u^*_{\mathcal{P}^{-X_k}}(\delta) = \max_{x_{\bar{k}}} \min\big(\phi^*_k\big(x_{\bar{k}}, \delta(x_{\bar{k}})\big), \psi^*_k\big(x_{\bar{k}}, \delta(x_{\bar{k}})\big), \psi^{-X_k}\big(x_{\bar{k}}, \delta(x_{\bar{k}})\big)\big) = u^*_{\mathcal{P}}(\delta). \qquad \square$$

Then, obviously the next corollary holds, since optimal strategies for $\mathcal{P}$ and $\mathcal{P}^{-X_k}$ should be identical:

Corollary 1. *An optimal optimistic strategy δ^* for $\mathcal{P}$ can be obtained by solving $\mathcal{P}^{-X_k}$ instead.*

Example 8. To show the application of the elimination of a state variable in the optimistic case, let us consider the omelette example.

First, it is important to recall that the variables will be eliminated with respect to the DAG-compatible ordering. In our example, this means that the variables in $\{C, F, O, S\}$ have to be eliminated first.

Let us start by eliminating state variable S: Here $X_k = S$.

$$\Phi_S = \pi_S, \qquad \overline{\Phi_S} = \{\pi_C, \pi_{OF}, \pi_O, \pi_F\},$$
$$\Psi_S = \{\mu_S\}, \qquad \overline{\Psi_S} = \{\mu_O, \mu_C\} \quad \text{and}$$
$$\psi^{-S}(F, PIO) = \max_{s \in S} \min\big\{\pi_S(s | F, PIO), \mu_S(s)\big\}.$$

Note that ψ^{-S} here only depends on $\{F, PIO\}$ and on no other variables. A new PID $\mathcal{P}^{-X_k} = (\mathcal{X}', \mathcal{D}', \mathcal{V}', \Phi', \Psi')$ can be computed.

$$\mathcal{X}' = \mathcal{X} - \{X_k\} = \{C, F, O, OF\},$$
$$\mathcal{D}' = \mathcal{D} = \{BAC, PIO\},$$
$$\mathcal{V}' = \mathcal{V} \cup \{V_{X_k}\} \backslash \{V_j | \mu_{V_j} \in \Psi_{X_k}\} = \{V_O, V_C, V_S\} \cup \{V'_S\} \backslash \{V_S\} = \{V_O, V_C, V'_S\}.$$

We obtain the PID of Fig. 7.

The possibility tables associated with this PID are $\pi_C(C|BAC)$, $\pi_{OF}(OF|F, BAC)$, $\pi_O(O|F, PIO)$ and $\pi_F(F)$; the initial possibility table $\pi_S(S|F, PIO)$ has been eliminated. The utility tables are $\mu_O(O)$, $\mu_C(C)$ and $\mu'_S(F, PIO)$; the initial utility table $\mu_S(S)$ has been eliminated.

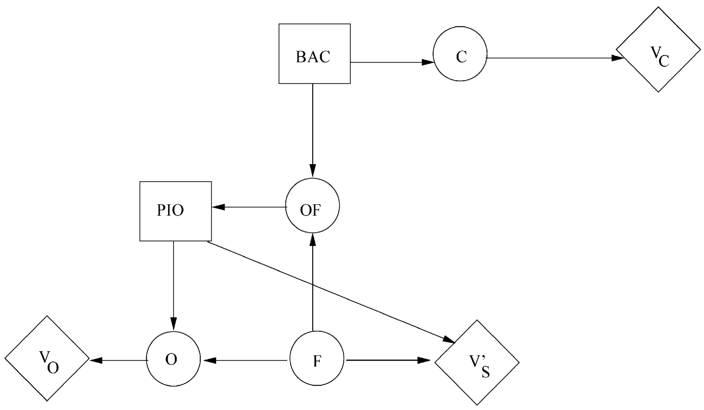

Fig. 7. PID after eliminating the variable S.

The new utility table $\mu'_S(F, PIO)$ is defined by

F, PIO	$\mu'_S(F, PIO)$
t, y	5
t, n	2
f, y	5
f, n	5

5.2.2. Pessimistic case

A similar state variable elimination method can also be defined in the pessimistic case. However in this case, a specific term ξ^{-X_k} is defined, instead of ψ^{-X_k} for variable $X_k \in \mathcal{X}_{end}$ which is not parent of any state variable.

Namely, ξ^{-X_k} is defined by:

$$\begin{aligned}\xi^{-X_k}(x_{\bar{k}}, d) = \min_{x_k \in X_k} \max\Big(&\max_{\pi_{X_i} \in \Phi_{X_k}} n\big(\pi_{X_i}\big(x_i | x_{Par(X_i)}, d_{Par(X_i)}\big)\big),\\ &\min\Big(\psi_k^*(x_{\bar{k}}, d), \min_{\mu_{V_j} \in \Psi_{X_k}} \mu_{V_j}\big(x_{Par(V_j)}, d_{Par(V_j)}\big)\Big)\Big), \quad \forall x, d.\end{aligned} \tag{11}$$

As for ψ^{-X_k}, ξ^{-X_k} may only depend on a subset of variables. Note that the main difference with the optimistic case is that ψ_k^* is included in the definition of ξ^{-X_k} contrarily to ψ^{-X_k}. This is due to the different form of alternation of max and min operators which, in the pessimistic case, prevents to extract ψ_k^*.

The following proposition can be shown:

Proposition 6.

$$u_*(\delta) = \min_{x_1,\ldots,x_{k-1},x_{k+1},\ldots,x_n} \max\big(n\big(\phi_k^*\big(x_{\bar{k}}, \delta(x_{\bar{k}})\big)\big), \xi^{-X_k}\big(x_{\bar{k}}, \delta(x_{\bar{k}})\big)\big).$$

Proof. Once again, if d^x denotes $\delta(x_{\bar{k}})$, we have:

$$\begin{aligned}
u_*(\delta) &= \min_{x_1,\ldots,x_n} \max\big(n\big(\pi_\delta(x_1,\ldots,x_n)\big), \mu_\delta(x_1,\ldots,x_n)\big)\\
u_*(\delta) &= \min_{x_1,\ldots,x_n} \max\Big(n\Big(\min_{\pi_{X_i} \in \Phi} \pi_{X_i}(x_i|x_{Par(X_i)}, d^x_{Par(X_i)})\Big), \min_{\mu_{V_j} \in \Psi} \mu_{V_j}(x_{Par(V_j)}, d^x_{Par(V_j)})\Big)\\
u_*(\delta) &= \min_{x_1,\ldots,x_n} \max\Big(n\Big(\min_{\pi_{X_i} \in \overline{\Phi_{X_k}}} \pi_{X_i}(x_i|x_{Par(X_i)}, d^x_{Par(X_i)})\Big), \max_{\pi_{X_i} \in \Phi_{X_k}} n\big(\pi_{X_i}(x_i|x_{Par(X_i)}, d^x_{Par(X_i)})\big),\\
&\qquad \min\Big(\min_{\mu_{V_j} \in \overline{\Psi_{X_k}}} \mu_{V_j}(x_{Par(V_j)}, d^x_{Par(V_j)}), \min_{\mu_{V_j} \in \Psi_{X_k}} \mu_{V_j}(x_{Par(V_j)}, d^x_{Par(V_j)})\Big)\Big)\\
u_*(\delta) &= \min_{x_1,\ldots,x_{k-1},x_{k+1},\ldots,x_n} \max\Big(n\big(\phi_k^*\big(x_{\bar{k}}, \delta(x_{\bar{k}})\big)\big), \min_{x_k} \max\Big(\max_{\pi_{X_i} \in \Phi_{X_k}} n\big(\pi_{X_i}(x_i|x_{Par(X_i)}, d^x_{Par(X_i)})\big),\\
&\qquad \min\Big(\psi_k^*\big(x_{\bar{k}}, \delta(x_{\bar{k}})\big), \min_{\mu_{V_j} \in \Psi_{X_k}} \mu_{V_j}(x_{Par(V_j)}, d^x_{Par(V_j)})\Big)\Big)\Big)\\
u_*(\delta) &= \min_{x_{\bar{k}}} \max\big(n\big(\phi_k^*\big(x_{\bar{k}}, \delta(x_{\bar{k}})\big)\big), \xi^{-X_k}\big(x_{\bar{k}}, \delta(x_{\bar{k}})\big)\big). \qquad \square
\end{aligned}$$

As in the optimistic case, we can show that the pessimistic utility of any strategy δ is unchanged in a PID when a chance variable belonging to the maximal class with respect to $\prec$ is eliminated. The only difference with the optimistic case is that in the modified PID $\mathcal{P}^{-X_k}$, Ψ is replaced with $\Psi' = \{\xi^{-X_k}\}$ and $\mathcal{V}$ is replaced with $\mathcal{V}' = \{V_{X_k}\}$. A unique utility node thus replaces all the remaining ones. Then, the following proposition can be proved:

Proposition 7 *(Consistency of state variable elimination (pessimistic case)). Let $\mathcal{P} = (\mathcal{X}, \mathcal{D}, \mathcal{V}, \Phi, \Psi)$ be a PID and $X_k \in \mathcal{X}_{end}$. The PID $\mathcal{P}^{-X_k} = (\mathcal{X}', \mathcal{D}', \mathcal{V}', \Phi', \Psi')$ where $\Psi' = \{\xi^{-X_k}\}$ is* equivalent *to $\mathcal{P}$. In other terms, any strategy δ has the same possibilistic pessimistic utility in $\mathcal{P}$ or in $\mathcal{P}^{-X_k}$.*

Proof. Let $u_*^{\mathcal{P}}(\delta)$ and $u_*^{\mathcal{P}^{-X_k}}(\delta)$ denote respectively the pessimistic utility of δ in $\mathcal{P}$ and $\mathcal{P}^{-X_k}$. From Proposition 6, we have:

$$u_*^{\mathcal{P}}(\delta) = \min_{x_1,\dots,x_{k-1},x_{k+1},\dots,x_n} \max\bigl(n\bigl(\phi_k^*\bigl(x_{\bar{k}}, \delta(x_{\bar{k}})\bigr)\bigr), \xi^{-X_k}\bigl(x_{\bar{k}}, \delta(x_{\bar{k}})\bigr)\bigr).$$

Now, from the definition of $\mathcal{P}^{-X_k}$, we have:

$$u_*^{\mathcal{P}^{-X_k}}(\delta) = \min_{x_1,\dots,x_{k-1},x_{k+1},\dots,x_n} \max\bigl(n\bigl(\min_{\pi_{X_i}\in\Phi'} \pi_{X_i}(x_i|x_{Par(X_i)}, d^x_{Par(X_i)})\bigr), \min_{\mu_{V_j}\in\Psi'} \mu_{V_j}(x_{Par(V_j)}, d^x_{Par(V_j)})\bigr)$$

$$u_*^{\mathcal{P}^{-X_k}}(\delta) = \min_{x_{\bar{k}}} \max\bigl(n\bigl(\phi_k^*\bigl(x_{\bar{k}}, \delta(x_{\bar{k}})\bigr)\bigr), \xi^{-X_k}\bigl(x_{\bar{k}}, \delta(x_{,\min\bar{k}})\bigr)\bigr) = u_*^{\mathcal{P}}(\delta). \qquad \square$$

Then, the next corollary holds, since optimal pessimistic strategies for $\mathcal{P}$ and $\mathcal{P}^{-X_k}$ should be identical:

Corollary 2. *An optimal pessimistic strategy δ^* for $\mathcal{P}$ can be obtained by solving $\mathcal{P}^{-X_k}$ instead of $\mathcal{P}$.*

Note that, in the pessimistic case, the elimination of a state variable results in grouping together all utility nodes. Thus, the resulting PID $\mathcal{P}^{-X_k}$ may take exponentially more space to be expressed than $\mathcal{P}$. In the optimistic case, the favourable alternation of min and max operators allows to keep utility functions distinct during state variable elimination and thus, may avoid this space complexity explosion. However, even in the pessimistic case, the next example shows that this complexity explosion can be sometimes avoided. Namely, when utility functions $\mu_{V_j} \in \overline{\Psi_{X_k}}$ do not share any variable with any utility function $\mu_{V_l} \in \Psi_{X_k}$ or any transition function $\pi_{X_i} \in \Phi_{X_k}$, they can still persist as independent functions and not be grouped in $\mu_{V_{X_k}}$. Thus, the space complexity explosion can be partially avoided.

Example 9. Once again we eliminate variable S from the PID. We have: $\Phi_S = \pi_S$, $\overline{\Phi_S} = \{\pi_C, \pi_{OF}, \pi_O, \pi_F\}$ and $\Psi_S = \{\mu_S\}$, $\overline{\Psi_S} = \{\mu_O, \mu_C\}$.

From this, we compute:

$$\begin{aligned}
\xi^{-S}(F, O, C, PIO) &= \min_{s\in S} \max\bigl(n\bigl(\pi_S(s|F, PIO)\bigr) \min\bigl(\mu_O(O), \mu_C(C), \mu_S(s)\bigr)\bigr) \\
&= \min\bigl\{\max\bigl(\min\bigl(\mu_O(O), \mu_C(C)\bigr), \min_{s\in S}\bigl(n\bigl(\pi_S(s|F, PIO)\bigr)\bigr)\bigr), \\
&\quad \min_{s\in S} \max\bigl(n\bigl(\pi_S(s|F, PIO)\bigr), \mu_S(s)\bigr)\bigr\} \\
&= \min\bigl\{\mu_O(O), \mu_C(C), \min_{s\in S} \max\bigl(n\bigl(\pi_S(s|F, PIO)\bigr), \mu_S(s)\bigr)\bigr\}.
\end{aligned}$$

In the general case all utility functions are replaced by a unique (but with maybe large scope) one. However in this example, $\xi^{-S}(F, O, C, PIO)$ is the minimum of three functions: μ_O, μ_C and a third one of scope $\{F, PIO\}$ which we name $\mu_{-S}(F, PIO)$. This is due to the fact that μ_O and μ_C do not share any variable with π_S or μ_S.

Thus, the new equivalent PID still has three utility functions: Only $\mu_S(S)$ is replaced with

$$\mu_{-S}(F, PIO) = \min_{s\in S} \max\bigl(n\bigl(\pi_S(s|F, PIO)\bigr), \mu_S(s)\bigr).$$

5.3. Elimination of a decision variable

Once all state variables in $\mathcal{X}_{end}$ (the set of state variables which are not the predecessor of any decision variable) have been eliminated by repeated application of the state variable elimination procedure, the next variable to remove will be a decision variable (belonging to the set of decision variables of the resulting PID which do not have any successor). Let $\mathcal{P} = (\mathcal{X}, \mathcal{D}, \mathcal{V}, \Phi, \Psi)$ be the current PID. In a similar way as for state variables, $\mathcal{D}_{end}$ will denote the set of decision variables in $\mathcal{D}$ which do not have any successor node in $\mathcal{X} \cup \mathcal{D}$. We are going to present a method for eliminating a decision variable D_j in $\mathcal{D}_{end}$ and building a local component δ_j^* of the optimal strategy δ^*.

Assume that D_j is the decision variable in $\mathcal{D}_{end}$ that has to be eliminated. As for state variable elimination, we define:

$$\Psi_{D_j} = \bigl\{\mu_V\bigl(Par(V)\bigr) \in \Psi,\ D_j \in Par(V)\bigr\}.$$

Now, let us also define Δ_{D_j}, the set of state and decision variables involved in Ψ_{D_j}:

$$\Delta_{D_j} = \left(\bigcup_{\mu_V(Par(V)) \in \Psi_{D_j}} Par(V) \right) - \{D_j\}.$$

It is easy to check that Δ_{D_j} is the set of variables whose instantiation determines a *local* utility degree together with the instantiation of D_j.

Similarly to the optimistic state variable elimination case, we define:

$$\psi^{-D_j}(x_{\Delta_{D_j}}, d_{\Delta_{D_j}}) = \max_{d_j \in D_j} \min_{\mu_V \in \Psi_{D_j}} \mu_V(x_{Par(V)}, d_{Par(V)}).$$

Note that obviously ψ^{-D_j} does not depend on D_j.

Eliminating a decision variable amounts to compute a *local strategy* $\delta^*_{D_j}$ which determines the value of this decision variable as a function of the values of variables in Δ_{D_j} so as to maximise the local utility function ψ^{-D_j}. Namely,

$$\delta^*_{D_j}(x_{\Delta_{D_j}}, d_{\Delta_{D_j}}) = \arg\max_{d_j \in D_j} \min_{\mu_V \in \Psi_{D_j}} \mu_V(x_{Par(V)}, d_{Par(V)}).$$

Proposition 8 shows that the obtained local strategy is part of a globally optimal strategy.

Proposition 8 *(Optimality of $\delta^*_{D_j}$). Let $\mathcal{P} = (\mathcal{X}, \mathcal{D}, \mathcal{V}, \Phi, \Psi)$ be a PID, where $\mathcal{D} = \{D_1, \ldots, D_p\}$ and let $D_j \in \mathcal{D}_{end}$ be a decision variable without any successor variable. Let also $\delta = \{\delta_{D_1}, \ldots, \delta_{D_p}\}$ be any fixed strategy, and $\delta^*_{D_j}$ be the local strategy computed through elimination of D_j. Let us also define $\delta_{\overline{D_j}} = \{\delta_{D_1}, \ldots, \delta_{D_{j-1}}, \delta_{D_{j+1}}, \ldots, \delta_{D_p}\}$.*

Then, $\quad u^*(\delta_{\overline{D_j}}, \delta^*_{D_j}) \geqslant u^*(\delta)$ *and* $u_*(\delta_{\overline{D_j}}, \delta^*_{D_j}) \geqslant u_*(\delta)$.

Proof. Let us write $\delta' = \{\delta_{\overline{D_j}}, \delta^*_{D_j}\}$. From (5) it is easily checked that $\pi_{\delta'}(x) = \pi_\delta(x)$, $\forall x$, since it is assumed that D_j has no descendent in the DAG. Let us write for a given x, $d = \delta(x)$ and $d' = \delta'(x)$. So, $d'_i = d_i, \forall i \neq j$. We have

$$\min_{l \text{ s.t. } D_j \in Par(V_l)} \mu_l(x_{Par(V_l)}, d'_{Par(V_l)}) = \max_{(d''_j \in D_j)} \min_{(l \text{ s.t. } D_j \in Par(V_l))} \mu_l(x_{Par(V_l)}, d_{Par(V_l)\setminus\{j\}}, d''_j) \quad \text{and}$$

$$\max_{(d''_j \in D_j)} \min_{(l \text{ s.t. } D_j \in Par(V_l))} \mu_l(x_{Par(V_l)}, d_{Par(V_l)\setminus\{j\}}, d''_j) \geqslant \min_{l \text{ s.t. } D_j \in Par(V_l)} \mu_l(x_{Par(V_l)}, d_{Par(V_l)}).$$

Since $\mu_l(x_{Par(V_l)}, d_{Par(V_l)}) = \mu_l(x_{Par(V_l)}, d'_{Par(V_l)})$, $\forall l$ s.t. $D_j \notin Par(V_l)$, we get $\mu_{\delta'}(x) \geqslant \mu_\delta(x)$, $\forall x$. From which we get both $u^*(\delta') \geqslant u^*(\delta)$ and $u_*(\delta') \geqslant u_*(\delta)$. $\square$

Now, let us define, for any PID $\mathcal{P} = (\mathcal{X}, \mathcal{D}, \mathcal{V}, \Phi, \Psi)$, and any variable $D_j \in \mathcal{D}_{end}$, a new PID $\mathcal{P}^{-D_j} = (\mathcal{X}', \mathcal{D}', \mathcal{V}', \Phi', \Psi')$ obtained after eliminating D_j:

- $\mathcal{X}' = \mathcal{X}$, $\mathcal{D}' = \mathcal{D} - \{D_j\}$ and $\mathcal{V}' = \mathcal{V} \cup \{V_{D_j}\} - \{V_l, \mu_{V_l} \in \Psi_{D_j}\}$,[8]
- $\Phi' = \Phi$ and $\Psi' = (\Psi - \Psi_{D_j}) \cup \{\psi^{-D_j}\}$.

Then, the following proposition shows that an optimal strategy can be obtained as the union of $\delta^*_{D_j}$ and of an optimal strategy for $\mathcal{P}'$.

Proposition 9 *(Consistency of decision variable elimination). Let $\mathcal{P} = (\mathcal{X}, \mathcal{D}, \mathcal{V}, \Phi, \Psi)$ be a PID, where $\mathcal{D} = \{D_1, \ldots, D_p\}$ and let $D_j \in \mathcal{D}_{end}$ a decision variable without any successor (chance or decision) node. Let $\mathcal{P}^{-D_j} = (\mathcal{X}', \mathcal{D}', \mathcal{V}', \Phi', \Psi')$ be the PID obtained after eliminating D_j and let $\delta^*_{D_j}$ be the corresponding computed local strategy.*

If $\delta^*_{\overline{D_j}}$ denotes an optimal strategy for $\mathcal{P}^{-D_j}$ then, for either possibilistic criterion, strategy $\delta^* = \{\delta^*_{\overline{D_j}}, \delta^*_{D_j}\}$ is an optimal strategy for $\mathcal{P}$ for the corresponding criterion.

[8] Here, V_{D_j} represents an additional utility node to which function ψ^{-D_j} is attached.

Proof. Let us first consider the optimistic criterion $u^*_{\mathcal{P}}$. Let $\delta^*_{\overline{D_j}}$ be an optimal strategy for $\mathcal{P}^{-D_j}$. Obviously, it holds that:

$$u^*_{\mathcal{P}^{-D_j}}(\delta^*_{\overline{D_j}}) \geqslant u^*_{\mathcal{P}^{-D_j}}(\delta_{\overline{D_j}}), \quad \forall \delta_{\overline{D_j}}. \tag{12}$$

It can also be checked from the definition of $\mathcal{P}^{-D_j}$ that for all $\delta_{\overline{D_j}}$:

$$u^*_{\mathcal{P}}(\delta_{\overline{D_j}}, \delta^*_{D_j}) = \max_{x\in\mathcal{X}} \min\Big(\min_{\pi_{X_i}\in\Phi} \pi_{X_i}(x_i|x_{Par(X_i)}, d_{Par(X_i)}), \\ \min_{\mu_{V_l}\in(\Psi-\Psi_{D_j})} \mu_{V_l}(x_{Par(X_i)}, d_{Par(X_i)}), \psi^{-D_j}(x_{\Delta_{D_j}}, d_{\Delta_{D_j}})\Big) = u^*_{\mathcal{P}^{-D_j}}(\delta_{\overline{D_j}}). \tag{13}$$

Now, from Eqs. (12) and (13), we get

$$u^*_{\mathcal{P}}(\delta^*_{\overline{D_j}}, \delta^*_{D_j}) \geqslant u^*_{\mathcal{P}}(\delta_{\overline{D_j}}, \delta^*_{D_j}), \quad \forall \delta_{\overline{D_j}}.$$

But now, Proposition 8 entails that $u^*_{\mathcal{P}}(\delta_{\overline{D_j}}, \delta^*_{D_j}) \geqslant u^*_{\mathcal{P}}(\delta)$, $\forall \delta$. So, finally,

$$u^*_{\mathcal{P}}(\delta^*_{\overline{D_j}}, \delta^*_{D_j}) \geqslant u^*_{\mathcal{P}}(\delta), \quad \forall \delta.$$

Thus, $\{\delta^*_{\overline{D_j}}, \delta^*_{D_j}\}$ is an optimal optimistic strategy for $\mathcal{P}$.

Let us now consider the pessimistic criterion $u^{\mathcal{P}}_*$. Once again it is easily checked that by definition

$$u_*^{\mathcal{P}^{-D_j}}(\delta^*_{\overline{D_j}}) \geqslant u_*^{\mathcal{P}^{-D_j}}(\delta_{\overline{D_j}}), \quad \forall \delta_{\overline{D_j}}. \tag{14}$$

Now, similarly as for (13), we can show:

$$u^{\mathcal{P}}_*(\delta_{\overline{D_j}}, \delta^*_{D_j}) = \min_{x\in\mathcal{X}} \max\Big(n\Big(\min_{\pi_{X_i}\in\Phi} \pi_{X_i}(x_i|x_{Par(X_i)}), d_{Par(X_i)}\Big), \\ \min\Big(\min_{\mu_{V_l}\in(\Psi-\Psi_{D_j})} \mu_{V_l}(x_{Par(X_i)}, d_{Par(X_i)}), \psi^{-D_j}(x_{\Delta_{D_j}}, d_{\Delta_{D_j}})\Big)\Big) = u_*^{\mathcal{P}^{-D_j}}(\delta_{\overline{D_j}}).$$

From these two equations we get

$$u^{\mathcal{P}}_*(\delta^*_{\overline{D_j}}, \delta^*_{D_j}) \geqslant u^{\mathcal{P}}_*(\delta_{\overline{D_j}}, \delta^*_{D_j}), \quad \forall \delta_{\overline{D_j}}.$$

Again, from Proposition 8 we get that $u^{\mathcal{P}}_*(\delta_{\overline{D_j}}, \delta^*_{D_j}) \geqslant u^{\mathcal{P}}_*(\delta)$, $\forall \delta$. So, finally,

$$u^{\mathcal{P}}_*(\delta^*_{\overline{D_j}}, \delta^*_{D_j}) \geqslant u^{\mathcal{P}}_*(\delta), \quad \forall \delta.$$

Thus, $\{\delta^*_{\overline{D_j}}, \delta^*_{D_j}\}$ is also an optimal pessimistic strategy for $\mathcal{P}$. $\square$

5.4. Arc reversal

When the current variable to eliminate (with respect to the DAG-compatible ordering) is a state variable which is a parent of another state variable, the results of Section 5.2 no longer hold. In this case, it is necessary to reverse first the dependency between the two variables to make sure to remove a node which is parent of utility nodes only. This arc reversal operation will not concern chance nodes which are parents of a decision node since, in this case, the chance node will be candidate to be removed only after the decision node has been removed (due to the DAG-compatible ordering).

Technically, this arc reversal operation takes the following form. Let X and Y be two state variables such that $X \in Par(Y)$ and such that the only oriented path from X to Y is the edge $X \to Y$ (Y is called a non-transitive child of X). We can define:

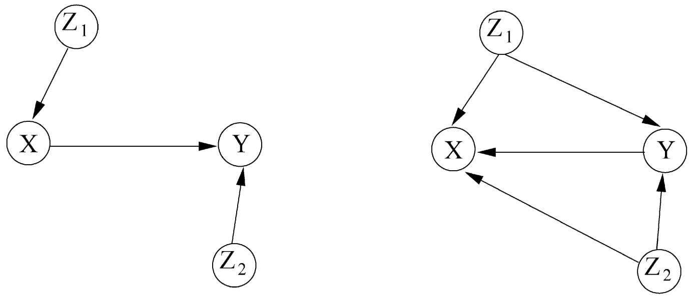

Fig. 8. Arc reversal operation.

$$\pi'_Y(Y|Par'(Y)) = \max_{x\in X} \min(\pi_Y(Y|Par(Y)), \pi_X(x|Par(X)))$$
$$\text{where } Par'(Y) = (Par(Y)\backslash\{X\}) \cup Par(X)$$
$$\pi'_X(X|Par'(X)) = \min(\pi_Y(Y|Par(Y)), \pi_X(X|Par(X))) * \pi'_Y(Y|Par'(Y))$$
$$\text{where } Par'(X) = Par(X) \cup \{Y\} \cup Par(Y)\backslash\{X\} \text{ and } a * b = a \text{ if } a < b \text{ and } a * b = 1_L \text{ if } a = b. \tag{15}$$

Fig. 8 shows how a link between X and Y is reversed by considering π'_X and π'_Y instead of π_X and π_Y. The consistency of arc reversal operation is shown by Proposition 10, which proof is omitted.

Proposition 10 *(Consistency of arc reversal). Let $\mathcal{P} = (\mathcal{X}, \mathcal{D}, \mathcal{V}, \Phi, \Psi)$ be a PID and let X and Y be two state variables such that Y is a non-transitive child of X and $X \in I_p$. Let $\mathcal{P}' = (\mathcal{X}, \mathcal{D}, \mathcal{V}, \Phi', \Psi)$, where $\Phi' = (\Phi'\backslash\{\pi_X(X|Par(X)), \pi_Y(Y|Par(Y))\}) \cup \{\pi'_X(X|Par'(X)), \pi'_Y(Y|Par'(Y))\}$, be the PID obtained after arc reversal. Let $\pi_\delta^{\mathcal{P}} = \min_{Z\in\mathcal{X}}(\pi_Z(Z|Par(Z)))$ and $\pi_\delta^{\mathcal{P}'} = \min_{Z\in\mathcal{X}}(\pi'_Z(Z|Par(Z)))$, then $\pi_\delta^{\mathcal{P}} = \pi_\delta^{\mathcal{P}'}$.*

Proposition 10 ensures that the joint possibility distribution π_δ is preserved by arc reversal operation from which $u^*(\delta)$ and $u_*(\delta)$ are also preserved. If it is necessary to remove a state variable X which has several children, it is then possible to find a sequence of arc reversal operations.

Example 10. In our example, it is necessary to reverse the edge between F and OF since, due to the partitioning of chance nodes, F should be eliminated before OF. $\pi_F(F)$ and $\pi_{OF}(OF|F, BAC)$ are known. After application of arc reversal, we get: $\pi'_{OF}(OF|BAC) = \max_{x\in F}(\min(\pi_{OF}(OF|F = x, BAC), \pi_F(F = x)))$ and $\pi'_F(F|OF, BAC) = \min(\pi_{OF}(OF|F, BAC), \pi_F(F)) * \pi'_{OF}(OF|BAC)$.

5.5. Variable elimination algorithm

Solving a PID simply consists in applying the state and decision variable elimination procedures to all variables in the PID, following a DAG-compatible ordering and applying arc reversal operations when necessary. The proof that the obtained strategy is optimal (according to the considered criterion) simply follows from Corollaries 1 and 2 and Propositions 9 and 10.

Example 11. Let us consider again the omelette example with the optimistic criterion. Once variable S has been eliminated, the state variable elimination process is repeated for variables C, O and F (following a DAG-compatible ordering). The PID of Fig. 9 results from the elimination of S, C and O.

A priori and conditional possibilities $\pi_F(F)$ and $\pi_{OF}(OF|BAC, F)$ do not change. Utilities $\mu'_C(BAC)$, $\mu'_S(PIO, F)$ and $\mu'_O(PIO, F)$ are:

BAC	$\mu'_C(BAC)$
y	4
n	5

PIO, F	$\mu'_S(PIO, F)$
y, t	5
y, f	5
n, t	2
n, f	5

PIO, F	$\mu'_O(PIO, F)$
y, t	5
y, f	0
n, t	3
n, f	3

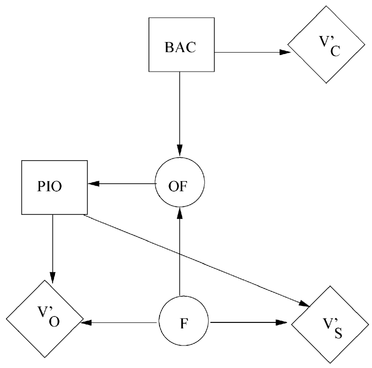

Fig. 9. PID obtained after elimination of variables S, C and O.

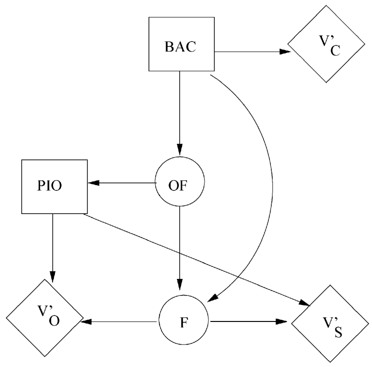

Fig. 10. PID after arc reversal between F and OF.

After the elimination of C and O, F has to be removed. To remove F, an arc reversal operation between F and OF has to be applied. The PID of Fig. 10 is obtained.

The new conditional possibilities $\pi'_{OF}(OF|BAC)$ and $\pi'_F(F|OF, BAC)$ are:

$\pi'_{OF}(OF\|BAC)$	y	n
t	5	0
f	3	0
u	0	5

$\pi'_F(F\|BAC, OF)$	(y,t)	(y,f)	(y,u)	(n,t)	(n,f)	(n,u)
t	5	0	5	5	5	5
f	0	5	5	5	5	3

Now, it is possible to remove node F. The PID of Fig. 11 is obtained.

The possibility table associated to this PID is $\pi'_{OF}(OF|BAC)$ and the utility tables are $\mu'_C(BAC)$ and $\mu'_F(OF, PIO, BAC)$. The new utility table defined by $\mu'_F(OF, PIO, BAC)$ is:

OF, PIO, BAC	$\mu'_F(OF, PIO, BAC)$	OF, PIO, BAC	$\mu'_F(OF, PIO, BAC)$
t, y, y	5	t, n, y	2
f, y, y	0	f, n, y	3
u, y, y	5	u, n, y	3
t, y, n	5	t, n, n	3
f, y, n	5	f, n, n	3
u, y, n	5	u, n, n	3

Now, variable PIO (which is a decision variable) has to be eliminated.

$$\mathcal{P}' = \big(\mathcal{X}, \mathcal{D} - \{PIO\}, \big(\mathcal{V} - \{V, \mu_V \in \Psi_{PIO}\}\big) \cup \{V_{PIO}\}, \Phi, (\Psi - \Psi_{PIO}) \cup \{\psi^{-PIO}\}\big).$$

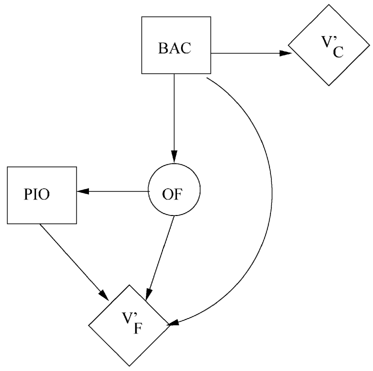

Fig. 11. PID obtained after elimination of variables S, C, O and F.

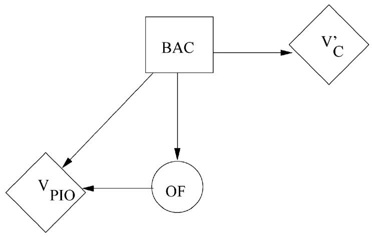

Fig. 12. PID obtained after elimination of variable PIO.

The PID obtained by elimination of PIO is the one of Fig. 12 since $\mathcal{X}' = \mathcal{X}$, $\mathcal{D}' = \mathcal{D} - \{PIO\} = \{BAC\}$ and $\mathcal{V}' = \mathcal{V} - \{V, \mu_V \in \Psi_{PIO}\} \cup V_{PIO} = \{V_{PIO}, V'_C\}$.

The possibility table $\pi'_{OF}(OF|BAC)$ and the two utility tables $\mu'_C(BAC)$ and $\mu_{PIO}(OF, BAC)$ are associated to the resulting PID. The last one (and its related local optimal strategy) is computed and gives:

OF, BAC	$\mu_{PIO}(OF, BAC)$	$\delta^*_{PIO}(OF, BAC)$
t, y	5	*yes*
f, y	3	*no*
u, y	5	*yes*
t, n	5	*yes*
f, n	5	*yes*
u, n	5	*yes*

The local "optimal" strategy is then $\delta^*_{PIO}(OF = \mathit{false}, BAC = \mathit{yes}) = \mathit{no}$ and $\delta^*_{PIO}(\mathit{other}) = \mathit{yes}$.

The algorithm is applied until no chance or decision node remains. Following the DAG-compatible ordering, state variable OF is removed after PIO and decision variable BAC is removed last.

The computation of the local strategy δ^*_{BAC} (which is unconditional) gives $\delta^*_{BAC} = \mathit{no}$.

The final PID is $\mathcal{P}$ such that $\mathcal{X} = \emptyset$, $\mathcal{D} = \emptyset$, $\mathcal{V}' = \{V_{BAC}\}$ and $\mu_{BAC} = 5$, where μ_{BAC} is the utility of the optimal optimistic strategy. The corresponding optimal strategy has been computed:

- $\delta^*_{PIO}(OF = \mathit{false}, BAC = \mathit{yes}) = \mathit{no}$ and $\delta^*_{PIO}(\mathit{other}) = \mathit{yes}$.
- $\delta^*_{BAC} = \mathit{no}$.

In other words, for this particular example and for this optimisation criterion, we have computed the sequential decisions $\delta^*_{BAC} = \mathit{no}$ and $\delta^*_{PIO} = \mathit{yes}$, which agrees with the intuition since, from an optimistic point of view, the fact that we think that the egg is fresh rather than rotten leads us to put it into the omelette without checking it.

6. Concluding remarks and perspectives

In this article we have described *Possibilistic Influence Diagrams* (PID), which allow to model in a compact form problems of sequential decision making under qualitative uncertainty in the framework of possibility theory. We have described a decision-tree based solution method for PID and we have given computational complexity results for several questions related to PID. Then, we have proposed *variable elimination* algorithms for solving PID for both optimistic and pessimistic criteria. We have also given proofs of correctness for these algorithms.

[17] independently presented a general unified framework for representing and solving various decision problems (from weighted CSP to stochastic ID or possibilistic MDP). This framework is more general than ours but not dedicated specifically to PID and does not include computational complexity considerations, except that their general model, being able to model QBF problems, is trivially PSPACE-hard. However, some of the methods proposed in [18,19] could certainly help to further improve our own variable elimination algorithms.

We now plan to study structured *possibilistic Markov decision processes* for which specific PID representations should be defined, representing explicitly the time-structure of the process (in the way 2-DBN are used to model structured MDP in [3]). This would lead to define structured versions of the possibilistic *value iteration* and *policy iteration* algorithms proposed in [22].

Another perspective is to study the relation between PID and possibilistic propositional logic or Constrained Satisfaction Problems. More precisely, PID problems can be transformed into problems of reasoning with possibilistic logic bases (computation of a degree of consistency or implication) for which dedicated algorithms have been proposed in the literature [6]. These algorithms reason from alpha-cuts and use a limited number of calls to SAT solvers in order to solve possibilistic decision problems expressed in possibilistic logic.

Acknowledgements

The authors would like to thank Cédric Pralet for helpful discussions which contributed to the writing of Section 5 (Variable elimination algorithms) of this article.

References

[1] S. Benferhat, D. Dubois, L. Garcia, H. Prade, On the transformation between possibilistic logic bases and possibilistic causal networks, Int. J. Approx. Reason. 29 (2002) 135–173.
[2] C. Boutilier, T. Dean, S. Hanks, Decision theoretic planning: Structural assumptions and computational leverage, J. Artificial Intelligence Res. 11 (2000) 1–94.
[3] C. Boutilier, R. Dearden, M. Goldszmidt, Stochastic dynamic programming with factored representations, Artificial Intelligence 121 (1) (2000) 49–107.
[4] D. Dubois, H. Fargier, P. Perny, Qualitative decision theory with preference relations and comparative uncertainty: An axiomatic approach, Artificial Intelligence 148 (2003) 219–260.
[5] D. Dubois, H. Fargier, H. Prade, Possibility theory in constraint satisfaction problems: Handling priority, preference and uncertainty, Appl. Intelligence 6 (1996) 287–309.
[6] D. Dubois, D. Le Berre, H. Prade, R. Sabbadin, Possibilistic logic for modeling qualitative decision: Atms-based algorithms, Fund. Inform. 34 (1999) 1–30.
[7] D. Dubois, H. Prade, Possibility theory as a basis for qualitative decision theory, in: Proc. of IJCAI'95, 1995, pp. 1925–1930.
[8] D. Dubois, H. Prade, R. Sabbadin, Decision-theoretic foundations of qualitative possibility theory, European J. Oper. Res. 128 (2001) 459–478.
[9] H. Fargier, J. Lang, R. Sabbadin, Towards qualitative approaches to multi-stage decision making, Int. J. Approx. Reason. 19 (1998) 441–471.
[10] L. Garcia, R. Sabbadin, Possibilistic influence diagrams, in: Proc. of ECAI'06, 2006, pp. 372–376.
[11] R.A. Howard, J.E. Matheson, Influence diagrams, in: Readings on the Principles and Applications of Decision Analysis, 1984, pp. 719–762.
[12] F.V. Jensen, Bayesian Networks and Decision Graphs, Springer, 2001.
[13] A. Mackworth, Consistency in networks of relations, Artificial Intelligence 8 (1977) 99–118.
[14] C.H. Papadimitriou, Computational Complexity, Addison-Wesley Publishing Company, 1994.
[15] J. Pearl, Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference, Morgan Kaufmann, 1988.
[16] C. Pralet, G. Verfaillie, T. Schiex, Composite graphical models for reasoning about uncertainties, feasibilities and utilities, in: Proc. of CP'05—Workshop on Preferences and Soft Constraints, Sitges, Spain, 2005.
[17] C. Pralet, G. Verfaillie, T. Schiex, Decision with uncertainties feasibilities and utilities: Towards a unified algebraic framework, in: Proc. of ECAI'06, 2006, pp. 427–431.
[18] C. Pralet, G. Verfaillie, T. Schiex, Decomposition of multi-operator queries on semiring-based graphical models, in: Proc. of CP'06, Nantes, France, 2006, pp. 437–452.
[19] C. Pralet, G. Verfaillie, T. Schiex, From influence diagrams to multioperator cluster DAGs, in: Proc. of UAI'06, Cambridge, MA, USA, 2006.

[20] M.L. Puterman, Markov Decision Processes, John Wiley and Sons, 1994.
[21] H. Raiffa, Decision Analysis: Introductory Lectures on Choices under Uncertainty, Addison-Wesley, Reading, MA, 1968.
[22] R. Sabbadin, Possibilistic Markov decision processes, Engrg. Appl. Artificial Intelligence 14 (2001) 287–300.
[23] R.D. Shachter, Evaluating influence diagrams, Oper. Res. 34 (6) (1986) 871–882.