Decision with uncertainties, feasibilities, and utilities: towards a unified algebraic framework

Cédric Pralet[1,3] and **Gérard Verfaillie**[2] and **Thomas Schiex**[3]

Abstract. Several formalisms exist to express and solve decision problems. Each is designed to capture different kinds of knowledge: utilities expressing preferences, uncertainties on the environment, or feasibility constraints on the decisions, with a possible sequential aspect. Despite the fact that every framework relies on specific properties exploited by dedicated algorithms, these formalisms present interesting similarities.

In this paper, we show that it is possible to capture these similarities in a generic algebraic framework for sequential decision making with uncertainties, feasibilities, and utilities. This framework subsumes several existing approaches, from constraint satisfaction problems to quantified boolean formulas, Bayesian networks or possibilistic Markov decision processes. We introduce this framework using a toy example, increasingly sophisticated by uncertainties, feasibilities and possible observations. This leads to a formal definition of the framework together with dedicated queries representing usual decision problems. Generic algorithms for solving the considered queries should allow to both factorize existing algorithmic works and allow for cross-fertilization between the subsumed formalisms.

1 Introduction and notations

The notion of decision problems covers a large spectrum of problems, from pure utility optimization problems to problems involving uncertainties, possible unfeasibilities and partial observability. A large number of frameworks have therefore been proposed to model and solve such problems. Our aim here is to define a general framework based on graphical models (to capture locality of information and independence) enriched by an algebraic framework allowing to design algorithms and prove properties. All proofs are omitted for lack of space and available in [27].

In the following, $Dom(x)$ denotes the set of values a variable x may take. By extension, for a sequence of variables S, $Dom(S) = \prod_{x \in S} Dom(x)$. Given a set E, a *local function* L is a function $L : Dom(S(L)) \rightarrow E$, where $S(L)$ is the scope of L. The *elimination of a set of variables* S' from L with any given associative commutative operator *op* on E is defined as $(op_{S'} L)(A) = op_{A' \in Dom(S')} L(A.A')$ for any assignment A of $S - S'$. Boolean values are denoted t and f.

We start with a first basic example using just utilities.

Example *John faces three doors A, B, C. One of the doors hides a treasure, another a gangster. John can decide to open one door. The gangster will rob him 4,000€ but the treasure is worth 10,000€.*

[1] LAAS-CNRS, Toulouse, France cpralet@laas.fr
[2] ONERA, Toulouse, France gerard.verfaillie@onera.fr
[3] INRA, Castanet-Tolosan, France tschiex@toulouse.inra.fr

Modeling To compactly represent the environment and the decisions, we introduce three variables: (1) two *environment variables*: one for the gangster door (denoted ga), and one for the treasure door (tr); (2) one *decision variable* (do), representing the door John decides to open. Every variable has $\{A, B, C\}$ as domain. Decision variables are variables whose value is controlled by an agent. Otherwise they are environment variables.

Then, we need two local *utility functions* U_1, U_2 to represent utilities: (1) U_1 expresses that if John opens the gangster door, he must pay $4,000$€ (soft constraint $do = ga$, with utility degree $-4,000$€ if satisfied, and 0 otherwise); (2) U_2 expresses that if John opens the treasure door, he wins $10,000$€ (soft constraint $do = tr$, with utility degree $10,000$€ if satisfied, and 0 otherwise).

Associated query Which door John should open if he knows that the gangster is behind door A and that the treasure is behind door C (no uncertainties)? Obviously, he should open door C.

1.1 Adding uncertainties

In real problems, the environment may not be completely known: there may be uncertainties (here called *plausibilities*) as well as possible *observations* on this uncertain environment. We sophisticate our treasure quest problem to integrate such aspects.

Example *The treasure and the gangster are not behind the same door, and all situations are equiprobable. John is accompanied by Peter. Each of them can decide to listen in to door A, B, or C to try to detect the dog of the gangster. The probability of hearing something is* 0.8 *if one listens in to the gangster door,* 0.4 *for a door next to the gangster one, and* 0 *otherwise.*

Modeling To capture these new specifications, we define (1) four more variables: two decision variables li_J and li_P, with $\{A, B, C\}$ as domain, model the doors to which John and Peter listen in, and two environment variables he_J and he_P, with $\{yes, no\}$ as domain, model whether John and Peter hear the dog; (2) four local *plausibility functions*: $P_1 : ga \neq tr$ and $P_2 = 1/6$ model probability distribution on the gangster and treasure positions; $P_3 = P(he_J \,|\, li_J, ga)$ defines the probability that John hears something given the door to which he listens in and the gangster door; similarly, P_4 corresponds to $P(he_P \,|\, li_P, ga)$. Implicitly, the local plausibilities satisfy *normalization conditions*. First, as the treasure and the gangster are somewhere, $\sum_{ga,tr} P_1 \times P_2 = 1$. Then, as John and Peter hear something or not, $\sum_{he_J} P_3 = 1$ and $\sum_{he_P} P_4 = 1$.

Associated queries What are the decision rules that maximize the expected utility, if first Peter and John listen in, and then John decides to open a door knowing what has been heard?

A classical approach to answer such a query is to use a *decision tree*. In this tree, variables can be considered in the order $li_J \to li_P \to he_J \to he_P \to do \to ga \to tr$ (first, John and Peter choose a door to listen in, then they listen and depending on the listening, John decides which door to open; finally the gangster and the treasure are behind a given door with a certain probability). An internal node n in the tree corresponds to a variable x, and an edge in the tree is labeled with an assignment $x = a$ of the variable x associated with the node above. If x is an environment variable, this edge is weighted by the probability $P(x = a \mid A)$, where A is the assignment corresponding to the path from the root to x.

The utility of a leaf node is the global utility $(U_1 + U_2)(A)$ of the complete assignment A associated with it. The utility of an internal decision node is given by the value of an optimal children (and it is possible to record an associated optimal decision). The utility of an internal environment node is given by the probabilistic expected utility of the values of its children nodes. The global expected utility is the utility of the root node. It can be proved [27] that such a decision tree procedure can be reduced to the computation of

$$\max_{li_J, li_P} \sum_{he_J, he_P} \max_{do} \sum_{ga,tr} ((\prod_{i\in[1,4]} P_i) \times (\sum_{i\in[1,2]} U_i))$$

In other words, the decision tree procedure is equivalent to a sequence of *variable eliminations* on a *combination* of *local functions*. Optimal decision rules can be recorded during the computation.

Different elimination sequences capture different problems or situations: if John thinks that Peter is a traitor and let him choose a door to listen in first (pessimistic attitude concerning the other agent), the sequence of elimination $\min_{li_P} \max_{li_J} \sum_{he_J, he_P} \max_{do} \sum_{ga,tr}$ is adequate (li_P is eliminated with $\min$). If one assumes that Peter does not even tell John what he has heard (John does not observe he_P), then the sequence of elimination becomes $\min_{li_P} \max_{li_J} \sum_{he_J} \max_{do} \sum_{he_P} \sum_{ga,tr}$.

1.2 Adding feasibilities

In some cases, conditions may have to be satisfied for a decision to be feasible. Unfeasibility is *not* infinite negative utility: an adversary cannot reach the former but seeks the latter.

Example *John and Peter cannot eavesdrop to the same door and door A is locked.*

Modeling This is achieved using two local *feasibility functions*: $F_1 : li_J \neq li_P$ and $F_2 : do \neq A$. We assume that at least one decision is feasible in any situation (no dead-end). It can be represented with two normalization conditions on feasibilities: $\vee_{li_J, li_P} F_1 = t$ and $\vee_{do} F_2 = t$. A decision tree procedure to answer queries is then equivalent to compute

$$\min_{li_P} \max_{li_J} \sum_{he_J} \max_{do} \sum_{he_P} \sum_{ga,tr} ((\bigwedge_{i\in[1,2]} F_i) \star (\prod_{i\in[1,4]} P_i) \times (\sum_{i\in[1,2]} U_i))$$

which uses a special operator $\star$ for masking unfeasible decisions: if we denote the utility of unfeasible situations by $\diamond$, then $\diamond$ should be an identity for elimination operators ($op(e, \diamond) = e$ for any elimination operator *op*), and annihilator for combination operators ($\diamond \otimes e = \diamond$ for any combination operator $\otimes$) since the combination of an unfeasible decision with other decisions is unfeasible. Then $\star$ is the operator that associates $\diamond$ with unfeasible decisions: $\star$ is defined by $f \star \alpha = \diamond$ and $t \star \alpha = \alpha$. Together, $\diamond$ and $\star$ exclude unfeasible situations from elimination domains.

Globally, the knowledge modeled with variables and local functions forms a *composite* graphical model defined by a DAG capturing normalization conditions on plausibilities and feasibilities (Figure 1(a))[4], and a network of local functions (Figure 1(b)). The network involves several types of variables (decision and environment variables) and several types of local functions (local utility, plausibility, and feasibility functions).

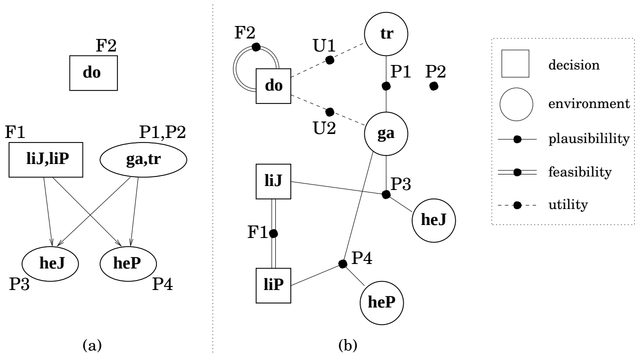

Figure 1. Composite graphical model (a) DAG capturing normalization conditions; (b) Network of local functions.

1.3 Other operators for plausibilities and utilities

The previous problem uses probabilities to model uncertainties. Under independence hypothesis, probabilities are combined with an operator $\otimes_p = \times$, and are eliminated (marginalized) with an operator $\oplus_p = +$. However, other uncertainty theories, such as possibility degrees [11] or κ-rankings [33], use other instantiations of $\otimes_p$/$\oplus_p$

Utilities considered so far are additive (using $\otimes_u = +$). However, utilities may express preferences combined with $\otimes_u = \min$.

Probabilistic expected utility is defined as $\sum_i p_i \times u_i$: plausibilities and utilities are combined with $\otimes_{pu} = \times$ and elimination on utilities weighted by plausibilities is done with $\oplus_u = +$. Other formalisms use an expected utility defined as $\underset{i}{\oplus_u} (p_i \otimes_{pu} u_i)$ using other instantiations of $\oplus_u$ and $\otimes_{pu}$:

	E_p	$\oplus_p$	$\otimes_p$	E_u	$\otimes_u$	$\oplus_u$	$\otimes_{pu}$
1	$\mathbb{R}^+$	$+$	$\times$	$\mathbb{R} \cup \{-\infty\}$	$+$	$+$	$\times$
2	$\mathbb{R}^+$	$+$	$\times$	$\mathbb{R}^+$	$\times$	$+$	$\times$
3	$[0,1]$	max	min	$[0,1]$	min	max	min
4	$[0,1]$	max	min	$[0,1]$	min	min	$\max(1{-}p, u)$
5	$\mathbb{N} \cup \{\infty\}$	min	$+$	$\mathbb{N} \cup \{\infty\}$	$+$	min	$+$

Table 1. Operators with - 1. probabilistic expected utility - 2. probabilistic expected satisfaction - 3. optimistic and 4. pessimistic possibilistic expected utility - 5. qualitative utility with κ-rankings and with only positive utilities.

Towards a generic framework After this informal introduction, it is now possible to give a formal definition of the *Plausibility-Feasibility-Utility* (PFU) framework. This framework combines graphical models concepts (locality, conditional independence) with a flexible algebraic definition of expected utility in an *algebraic graphical model* (Section 2). We then show how queries on these networks allow to solve decision problems, with or without partial observability.

Ideally, such queries should reduce to a sequence of variable eliminations on a combination of local functions such as

$$\max_{x_1,x_2} \min_{x_3} \oplus_{u\, x_4,x_5} \max_{x_6} ((\bigwedge_{F_i\in F} F_i) \star (\bigotimes_{p\, P_i\in P} P_i) \otimes_{pu} (\bigotimes_{u\, U_i\in U} U_i))$$

[4] If P denotes the set of local plausibility functions associated with a node corresponding to a set of variables S, this means that $\sum_S (\prod_{P_i\in P} P_i) = 1$. If F denotes the set of local feasibility functions associated with a node corresponding to a set of variable S, this means that $\vee_S(\wedge_{F_i\in F} F_i) = t$.

2 An algebraic graphical model

2.1 Algebraic structure (plausibility/utility model)

Plausibility structure We define a *plausibility structure* as a triple $(E_p, \oplus_p, \otimes_p)$, where E_p is a set of plausibility degrees, equipped with a partial order $\preceq_p$, $\oplus_p$ is an elimination operator on plausibilities, and $\otimes_p$ is a combination operator on plausibilities. $\oplus_p$ and $\otimes_p$ satisfy some sensible axioms inspired by Friedman & Halpern's algebraic plausibility measures [14, 17]. The difference with their work is that we extend the operators $\oplus_p$ and $\otimes_p$ so that they become closed in E_p. This allows to define $(E_p, \oplus_p, \otimes_p)$ as a commutative semiring (the identity for $\oplus_p$ is noted 0_p and the identity for $\otimes_p$ is noted 1_p), where $\oplus_p$ and $\otimes_p$ are monotonic, and where 0_p (associated with impossibility) is the minimum element in E_p.

Utility structure We define a *utility structure* as a pair $(E_u, \otimes_u)$, where E_u is a set of utility degrees, (equipped with a partial order $\preceq_u$) and $\otimes_u$ is the operator used to combine utilities. We assume that $\otimes_u$ is monotonic and that $(E_u, \otimes_u)$ is a commutative semi-group ($\otimes_u$ associative and commutative, so that combined utilities do not depend on the way the combination is done).

Expected utility structure To simultaneously take into account plausibilities and utilities, we define an *expected utility structure*. This structure, inspired from Chu-Halpern's work on generalized expected utility [8] (as in algebraic MDP [26]), is a tuple $(E_p, E_u, \oplus_u, \otimes_{pu})$ where $\oplus_u$ (operator on E_u) together with $\otimes_{pu} : E_p \times E_u \to E_u$ define the generalized expected utility formula $\underset{i}{\oplus_u} (p_i \otimes_{pu} u_i)$. We extend [8] in order to be able to deal with *sequential* decision processes. For operational reasons, $\oplus_u$ and $\otimes_{pu}$ are also closed. This defines $(E_p, E_u, \oplus_u, \otimes_{pu})$ as a semi-module on $(E_p, \oplus_p, \otimes_p)$, with monotonicity axioms on $\oplus_u$ and $\otimes_{pu}$. Note that all the structures in Table 1 are expected utility structures.

Implicit assumptions As a set E_p of plausibility degrees and a set E_u of utility degrees are defined, plausibilities and utilities must be cardinal: purely ordinal approaches (e.g. CP-nets [4]) are not captured. As $\otimes_{pu}$ takes values in E_u, it is implicitly assumed that plausibilities and utilities are commensurable: works as [13] are not captured either. Finally, some axioms entail that only distributional plausibilities are covered (the plausibility of a set of variable assignments is determined by the plausibilities of each covered complete assignment): belief functions [30] are not captured.

Algebraic structure for the treasure problem The plausibility structure is $(\mathbb{R}^+, +, \times)$ ($E_p = \mathbb{R}^+$ and not $[0, 1]$ in order for $+$ to be closed in E_p), the utility structure is $(\mathbb{R}, +)$, and the expected utility structure is $(\mathbb{R}^+, \mathbb{R}, +, \times)$.

2.2 A generic graphical model

Following the graphical model concepts, the environment and the decisions are now represented by environment and decision variables while the relations between these variables (plausibility, feasibility, and utility relations) are represented as local functions. Together, these define a composite and generic graphical model called a *Plausibility-Feasibility-Utility (PFU) network*.

Definition 1 *A PFU network is a tuple (V, G, P, F, U) where:*

- *V is a set of variables, partitioned into V_D (set of decision variables) and V_E (set of environment variables).*
- *G is a DAG whose vertices are a partition of V (vertices of G are sets of variables called* components*), such that each vertex of G is composed of variables of the same nature (decision or environment). We note $\mathcal{C}_D$ the set of decision components and $\mathcal{C}_E$ the set of environment components.*
- *$P = \{P_1, P_2, \ldots\}$ is a finite set of local* plausibility functions*; each P_i is associated with a component $c \in \mathcal{C}_E$ denoted $c(P_i)$, such that the scope $S(P_i) \subset (c \cup pa_G(c))$. For any $c \in \mathcal{C}_E$. $\underset{c}{\oplus_p}(\otimes_{p\, c(P_i)=c} P_i) = 1_p$ must hold.*
- *$F = \{F_1, F_2, \ldots\}$ is a finite set of local* feasibility functions*; each F_i is associated with a component $c \in \mathcal{C}_D$ denoted $c(F_i)$, such that the scope $S(F_i) \subset (c \cup pa_G(c))$. For any $c \in \mathcal{C}_D$, we have $\underset{c}{\vee}(\wedge_{c(F_i)=c} F_i) = t$.*
- *$U = \{U_1, U_2, \ldots\}$ is a finite set local* utility functions*.*

The decomposition of global plausibility, feasibility, and utility degrees as sets of local functions is semantically justified by the notion of *conditional independence*. We provide an intuitive justification for plausibilities. Let us say that $\mathcal{P}_S$ is a plausibility distribution on S iff $\oplus_{p\, S} \mathcal{P}_S = 1_p$ (for probabilities, it simply means that a probability distribution sums up to 1). $\mathcal{P}_S$ defines a plausibility distribution on any subset S' of S by $\mathcal{P}_{S'} = \oplus_{p\, S-S'} \mathcal{P}_S$.

To define conditional independence, we introduce a conditioning function $\oslash_p$ (verifying some sensible properties, see [27]) allowing to define conditional plausibility distributions by $\mathcal{P}_{S_1 \mid S_2} = \mathcal{P}_{S_1, S_2} \oslash_p \mathcal{P}_{S_2}$ (for probabilities, $\oslash_p$ is the division). Then, for any disjoint sets of variables S_1, S_2, S_3, we say that S_1 is *conditionally independent* of S_2 given S_3 iff $\mathcal{P}_{S_1, S_2 \mid S_3} = \mathcal{P}_{S_1 \mid S_3} \otimes_p \mathcal{P}_{S_2 \mid S_3}$.

Last, a DAG G of components is said to be *compatible* with a plausibility distribution $\mathcal{P}_S$ iff for any component c of the DAG, c is conditionally independent of its non descendants in G given the set $pa_G(c)$ of its parents in G. Similarly to Bayesian networks [25], it is possible to prove [27] that:

Theorem 1 *If G is a DAG compatible with $\mathcal{P}_S$, then $\mathcal{P}_S = \otimes_{p\, c \in G} \mathcal{P}_{c \mid pa_G(c)}$.*

Thanks to Theorem 1[5], a DAG compatible with a global plausibility distribution enables to factorize it as a combination of local functions $L_{c,pa_G(c)}$ ($L_{c,pa_G(c)} = \mathcal{P}_{c \mid pa_G(c)}$) such that $\oplus_{p\, c} L_{c,pa_G(c)} = 1_p$ for any component c of G. Each $L_{c,pa_G(c)}$ may be further decomposed to give local plausibility functions P_i associated with c (denoted $c(P_i) = c$) thus verifying $\oplus_{p\, c} (\otimes_{p\, c(P_i)=c} P_i) = 1_p$. This two-step factorization justifies the DAG in a PFU network and the normalization conditions it encodes.

Similar results can be established for feasibilities. As for utilities, PFU networks implicitly assume that a global utility degree $\mathcal{U}_V$ on all variables can be decomposed as a set of local utility functions U_i. One may assume that this decomposition is directly obtained, as it is done with CSP [22] or is justified by a notion of conditional independence in the case $\otimes_u = +$ as in [1].

Example The PFU network of our example appears in Figure 1.

Subsumption results CSP [22] can be modeled as $(V, G, \emptyset, \emptyset, U)$ where V contains the CSP variables, the DAG G is reduced to one decision component equal to V and U is the set of constraints (by normalization, feasibilities cannot represent inconsistent CSP).

[5] And with some technical steps induced by the fact that we do not work on plausibility distributions on V, but on plausibility distributions on V_E for any assignment of V_D.

It is also possible to capture valued [2], quantified [3], mixed [12], stochastic [34] CSP, or SAT, QBF, and extended stochastic SAT [21].

Bayesian networks [25] are captured by a PFU network $(V, G, P, \emptyset, \emptyset)$, where P contains the original probability tables. Chain graphs [15], Markov random fields [7] are also subsumed.

Finite-horizon probabilistic MDP [28, 23] are captured by $(V, G, P, \emptyset, U)$ where V_D and V_E are the set of decisions d_t and states $s - t$ respectively (one for each time-step t), G is a DAG looking like the unrolled MDP, P contains the probability distributions $P_{s_{t+1} \mid s_t, d_t}$, and U contains the additive rewards R_{s_t, d_t}. It is also possible to model finite-horizon possibilistic MDP [29], MDP based on κ-rankings, partially observable (PO) MDP, factored or not [5, 6].

Influence diagrams [18], including a possibilistic variant, can be represented as a tuple $(V, G, P, \emptyset, U)$. For valuation networks [32], a set F of local feasibility functions is added.

3 Reasoning about PFU networks via queries

In this section, we assume that a *sequence of decisions* must be performed, and that the order in which decisions are made and the environment is observed is known. We also make a *no-forgetting* assumption, that is, when making a decision, an agent is aware of all previous decisions and observations. Finally, the order on utility degrees is assumed to be total.

Under such assumptions, we want to express sequential decision making problems on PFU networks, taking into account possible partial observability and cooperative or antagonist agents in the environment. To capture such problems, we use a sequence Sov of operator-variable(s) pairs that captures:

- *possible unobservabilities*: the order in which decisions are made and environment variables observed is specified by Sov. If the value of he_P is known when John chooses a door to open, then Sov contains $\ldots(\oplus_u, he_P)\ldots(\max, do)\ldots$. Otherwise, a sequence like $\ldots(\max, do)\ldots(\oplus_u, he_P)\ldots$ is used;
- *optimistic/pessimistic attitude* concerning the decision makers: if Peter acts cooperatively (we are optimistic about Peter's decision) then $(\max, li_P)$ appears in Sov. If instead Peter is considered as an antagonist agent then $(\min, li_P)$ is used.

Example For the last query in §1.1, Sov equals $(\min, \{li_P\}).(\max, \{li_J\}).(+, \{he_J\}).(\max, \{do\}).(+, \{he_P\}). (+, \{ga, tr\})$.

Definition 2 *A query Q on a PFU network is a pair $(\mathcal{N}, Sov)$ where (1) $\mathcal{N}$ is a PFU network; (2) Sov is a sequence of* operator-variable(s) *pairs such that the operators are* min, max *or $\oplus_u$, and such that each variable appears at most once in Sov.*

Correct queries Not all queries are meaningful. The main condition for a query to be *correct* is that it must not contain a pair x, y of variables of different nature such that x belongs to an ascendant component of y in the DAG of the PFU network and x appears after y in Sov. This would mean that x is assigned after y, breaking causality. For example, the pair $\ldots(+, he_J)\ldots(\max, li_J)\ldots$ is not correct since John cannot hear something at the door he has chosen to eavesdrop before this choice is done.

3.1 Answering queries

Asnwering a query Q consists in computing the expected utility associated with the situation modeled by the sequence Sov and the network $\mathcal{N}$ of the query Q.

Decision tree approach A first approach to answer queries in the general case (not only for probabilistic expected utility) uses decision trees. In this case, variables are considered as they appear in Sov, and edges in the tree are weighted by conditional plausibilities of the form $\mathcal{P}(x = a \mid A)$ for the internal nodes associated with environment variables, and by conditional feasibilities of the form $\mathcal{F}(x = a \mid A)$ for internal nodes associated with decision variables.

Then the expected utility of a query (and associated optimal decision rules specifying which decision to take given the previous observed variables) can be defined with a decision tree procedure similar to the procedure described in 1.1 (the utility of a leaf node is given by the combination of the local utilities U_i, the utility of an internal environment node is given by the expected utility of its children, and the utility of an internal decision node is given by the optimal utility of its feasible children nodes).

A more operational approach The advantage of the decision tree procedure is that it has clear semantic foundations. But besides the possibly exponential size tree, its drawback is that each internal node of the decision tree may require the computation of $\mathcal{P}(x = a \mid A)$ or $\mathcal{F}(x = a \mid A)$ which are not usually directly available in the network $\mathcal{N}$ and which may require exponential time to compute. Fortunately, it is possible to show [27] that the decision tree procedure (called the semantic answer to Q) is equivalent to a direct algebraic approach (called the operational answer to Q) which requires only the local functions available in the original PFU network.

Theorem 2 *Answering a query with a decision tree is equivalent to compute $Ans(Q) = Sov((\bigwedge_{F_i \in F} F_i) \star (\bigotimes_{p\, P_i \in P} P_i) \otimes_{pu} (\bigotimes_{u\, U_i \in U} U_i))$ Furthermore, the optimal decision rules obtained are the same as in the decision tree approach.*

3.2 Subsumption of classical queries

Most usual queries on existing formalisms can be reduced to PFU queries: finding a solution for a SAT problem, a CSP [22] or a valued CSP [2] corresponds to a sequence $Sov = (\max, V)$. For QBF or quantified CSP [3], Sov alternates $\min$ (for universal quantification) and $\max$ (for existential quantification). With mixed or probabilistic CSP [12], Sov looks like $(\oplus_u, V_E).(\max, V_D)$ if a conditional decision is sought and $(\max, V_D).(\oplus_u, V_E)$ if an unconditional decision is sought. The situation is similar with conformant or probabilistic planning [16].

Queries on Bayesian networks [25] look like $(+, S)$ to compute a probability distribution on $V - S$, $(\max, V)$ to solve a Most Probable Explanation problem, and $(\max, V_D).(+, V_E)$ to solve Maximum A Posteriori problems.

With stochastic CSP [34] or influence diagrams [18], the sequence alternates $+$ on environment variables and $\max$ on decision variables. With finite-horizon MDP, Sov looks like $(\max, d_1).(\oplus_u, s_2).\ldots.(\max, d_T).(\oplus_u, s_T)$. With finite-horizon POMDP, observations o_t are added (one for each time-step t), and $Sov = (\max, d_1).(\oplus_u, o_2).\ldots.(\max, d_T).(\oplus_u, o_T).(\oplus_u, \{s_1, \ldots, s_T\})$: this captures the fact that states remains unobserved for POMDP.

4 Gains and costs

A better understanding As it subsumes many queries on existing graphical models, the PFU framework enables to better understand the similarities and differences between the subsumed formalisms.

It defines a common basis for people of different communities to communicate.

Increased expressiveness The PFU framework offers several variabilities: (1) variability of the algebraic structure, which captures probabilistic expected utility, probabilistic expected satisfaction, possibilistic pessimistic utility, possibilistic optimistic utility or qualitative utility with κ-rankings; (2) variability of the network which exploits oriented and non-oriented independences as well as normalization conditions; (3) variability of the queries which can capture state (un)observability and ccoperative/antagonist attitudes.

It is therefore more expressive than each of the frameworks it subsumes, and it also covers yet unpublished formalisms (such as possibilistic influence diagrams, or stochastic CSP extended to cope with the fact that decisions may influence the environment).

Generic algorithms Computing the answer to a correct query is obviously PSPACE-hard since PFU queries capture QBF. It is easy to define, from Theorem 2, a polynomial space tree search algorithm which computes the answer to a correct query. Similarly, it is possible to define a generic variable elimination algorithm to compute $Ans(Q)$ [10]. The PFU algebraic framework is an opportunity to identify sufficient or necessary conditions for existing algorithms to be applicable [24], or to define new techniques from which each subsumed formalism could benefit. Bounding and local consistencies [22, 9, 20] could be integrated to speed up the resolution.

As a result, the PFU framework can be seen as an opportunity to integrate in a generic framework techniques developed in different subsumed formalisms, and thus to allow for cross-fertilization.

5 Conclusion

In this paper[6], a generic algebraic framework for sequential decision making has been defined. It combines an algebraic structure that specifies how to combine and synthesize information together with a graphical model specifying local plausibility, feasibility, and utility functions. Queries can capture possible (un)observabilities or antagonist agents.

The generalized expected utility associated with a query can be computed by a sequence of variable eliminations on a combination of local functions. Compared to *valuation algebras* [31, 19], a related generic framework, the PFU framework uses several combination ($\wedge$, $\star$, $\otimes_p$, $\otimes_{pu}$, $\otimes_u$) and elimination operators ($\min$, $\max$, $\oplus_u$). Moreover, the semantic justifications of the definition of PFU networks, which lie in the notion of conditional independence, allow to include a DAG capturing normalization conditions in the network definition.

The obtained framework not only subsumes many queries on existing formalisms, but it also enables to define yet unpublished formalisms. From an algorithmic point of view, generic schemes that integrate techniques used in subsumed formalisms can be developed.

[6] This work was partially conducted within the EU IP COGNIRON ("The Cognitive Companion") funded by the European Commission Division FP6-IST Future and Emerging Technologies under Contract FP6-002020.

References

[1] F. Bacchus and A. Grove, 'Graphical Models for Preference and Utility', in *Proc. of UAI*, (1995).

[2] S. Bistarelli, U. Montanari, F. Rossi, T. Schiex, G. Verfaillie, and H. Fargier, 'Semiring-Based CSPs and Valued CSPs: Frameworks, Properties and Comparison', *Constraints*, **4**(3), 199–240, (1999).

[3] L. Bordeaux and E. Monfroy, 'Beyond NP: Arc-consistency for Quantified Constraints', in *Proc. of the 8th International CP conference*, Ithaca, New York, USA, (2002).

[4] C. Boutilier, R. Brafman, H. Hoos, and D. Poole, 'Reasoning With Conditional Ceteris Paribus Preference Statements', in *Proc. of the 15th UAI Conference*, Stockholm, Sweden, (1999).

[5] C. Boutilier, R. Dearden, and M. Goldszmidt, 'Stochastic Dynamic Programming with Factored Representations', *Artificial Intelligence*, **121**(1-2), 49–107, (2000).

[6] Craig Boutilier, Thomas Dean, and Steve Hanks, 'Decision-theoretic planning: Structural assumptions and computational leverage', *Journal of Artificial Intelligence Research*, **11**, 1–94, (1999).

[7] R. Chellappa and A. Jain. Markov random fields: Theory and applications. Academic Press, 1993.

[8] F.C. Chu and J.Y. Halpern, 'Great expectations. part i: On the customizability of generalized expected utility', in *Proc. of IJCAI*, pp. 291–296, Acapulco, Mexico, (2003).

[9] M. Cooper and T. Schiex, 'Arc Consistency for Soft Constraints', *Artificial Intelligence*, **154**(1-2), 199–227, (2004).

[10] R. Dechter, 'Bucket Elimination: A Unifying Framework for Reasoning', *Artificial Intelligence*, **113**(1-2), 41–85, (1999).

[11] D. Dubois and H. Prade. Possibility Theory: An Approach to Computerized Processing of Uncertainty. Plenum Press, 1988.

[12] H. Fargier, J. Lang, and T. Schiex, 'Mixed Constraint Satisfaction: a Framework for Decision Problems under Incomplete Knowledge', in *Proc. of the 13th AAAI*, pp. 175–180, Portland, USA, (1996).

[13] H. Fargier and P. Perny, 'Qualitative Models for Decision Under Uncertainty without the Commensurability Assumption', in *Proc. of UAI*, (1999).

[14] Friedman N., Halpern J., 'Plausibility Measures : A User's Guide', in *Proc. of UAI*, Montral, Canada, p. 175-184, (1995).

[15] M. Frydenberg, 'The Chain Graph Markov Property', *Scandinavian Journal of Statistics*, **17**, 333–353, (1990).

[16] M. Ghallab, D. Nau, and P. Traverso, *Automated Planning: Theory and Practice*, Morgan Kaufmann, 2004.

[17] J.Y. Halpern, 'Conditional plausibility measures and Bayesian networks', *JAIR*, **14**, 359–389, (2001).

[18] R. Howard and J. Matheson, 'Influence Diagrams', in *Readings on the Principles and Applications of Decision Analysis*, 721–762, Strategic Decisions Group, Menlo Park, CA, USA, (1984).

[19] J. Kolhas, *Information Algebras: Generic Structures for Inference*, Springer, 2003.

[20] J. Larrosa and T. Schiex., 'In the quest of the best form of local consistency for weighted CSP', in *Proc. of IJCAI*, Acapulco, Mexico, (2003).

[21] M. Littman, S. Majercik, and T. Pitassi, 'Stochastic Boolean Satisfiability', *Journal of Automated Reasoning*, **27**(3), 251–296, (2001).

[22] A. Mackworth, 'Consistency in Networks of Relations', *Artificial Intelligence*, **8**(1), 99–118, (1977).

[23] G. Monahan, 'A Survey of Partially Observable Markov Decision Processes: Theory, Models, and Algorithms', *Management Science*, **28**(1), 1–16, (1982).

[24] P. Ndilikilikesha, 'Potential Influence Diagrams',*International Journal of Approximated Reasoning*,**3**, 251–285, (1994).

[25] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann, 1988.

[26] P. Perny, O. Spanjaard, and P. Weng, 'Algebraic Markov Decision Processes', in *19th IJCAI Proc.*, Edinburgh, Scotland, (2005).

[27] C. Pralet, G. Verfaillie, and T. Schiex. An Algebraic Graphical Model for Decision with Uncertainties, Feasibilities, and Utilities. LAAS-CNRS Report, http://www.laas.fr/∼cpralet/praletverfschiex.ps, 2005.

[28] M. Puterman, *Markov Decision Processes, Discrete Stochastic Dynamic Programming*, John Wiley & Sons, 1994.

[29] R. Sabbadin, 'A Possibilistic Model for Qualitative Sequential Decision Problems under Uncertainty in Partially Observable Environments', in *Proc. of the 15th UAI Conference*, Stockholm, Sweden, (1999).

[30] G. Shafer, *A mathematical theory of evidence*, Princeton University Press, 1976.

[31] P. Shenoy, 'Valuation-based Systems for Discrete Optimization', *Uncertainty in Artificial Intelligence*, **6**, 385–400, (1991).

[32] P.P. Shenoy, 'Valuation network representation and solution of asymmetric decision problems', *European Journal of Operational Research*, **121**, 579–608, (2000).

[33] W. Spohn, 'A general non-probabilistic theory of inductive reasoning', in *Uncertainty in Artificial Intelligence 4*, 149–158, North-Holland, Amsterdam, (1990).

[34] T. Walsh, 'Stochastic Constraint Programming', in *Proc. of the 15th ECAI Conference*, Lyon, France, (2002).