

# Décision avec incertitudes, faisabilités et utilités : vers un cadre algébrique unifié

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

Cédric Pralet<sup>1</sup>

Gérard Verfaillie<sup>2</sup>

Thomas Schiex<sup>3</sup>

<sup>1</sup> LAAS-CNRS & INRA/BIA, Toulouse, cpralet@laas.fr

<sup>2</sup> ONERA/DCSD, Toulouse, Gerard.Verfaillie@onera.fr

<sup>3</sup> INRA/BIA, Toulouse, tschiex@toulouse.inra.fr

### Résumé

*Différents formalismes existent pour exprimer et résoudre des problèmes de décision. Chacun d'entre eux est conçu pour représenter différents types de connaissance : des utilités exprimant des préférences, des incertitudes sur l'environnement ou des contraintes de faisabilité sur les décisions, avec un éventuel aspect séquentiel. Bien que chaque formalisme repose sur des propriétés spécifiques exploitées par des algorithmes dédiés, ils présentent tous d'intéressantes similarités.*

*Dans ce papier, nous montrons qu'il est possible de capturer ces similarités dans un cadre algébrique générique pour la décision séquentielle avec incertitudes, faisabilités et utilités. Ce cadre inclut de nombreux formalismes existants, comme les problèmes de satisfaction de contraintes, les formules booléennes quantifiées, les réseaux bayésiens ou les processus décisionnels markoviens.*

*Nous l'introduisons grâce à un exemple jouet, incrémentalement sophistiqué par la prise en compte d'incertitudes, de possibilités d'observation et de faisabilités. Ceci nous conduit à la définition formelle du cadre proposé et des requêtes associées, exprimant des problèmes de décision classiques. Incidemment, ceci nous permet de définir des formalismes non encore identifiés, comme par exemple les diagrammes d'influence possibilistes. La définition d'algorithmes génériques pour répondre aux requêtes devrait favoriser le partage de connaissances entre communautés diverses et la factorisation des nombreux travaux algorithmiques existants dans ces communautés.*

### Mots Clef

Décision séquentielle, Incertitudes, Faisabilités, Utilités.

### 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 formalism relies on specific properties exploited by dedicated algorithms, all of them 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 formalisms such as constraint satisfaction problems, quantified boolean formulas, Bayesian networks, or Markov decision processes.*

*We introduce it using a toy example, increasingly sophisticated by adding uncertainties, possible observations, and feasibilities. This leads us to the formal definition of the proposed framework, together with associated queries representing usual decision problems. As a side effect, this actually defines yet unpublished formalisms, such as possibilistic influence diagrams. Designing generic algorithms for answering queries should encourage knowledge sharing between various communities and allow for cross-fertilization between algorithmic works existing in these communities.*

### Keywords

Sequential Decision-making, Uncertainties, Feasibilities, Utilities.

## 1 Introduction et notations

La notion de problème de décision couvre un large spectre de problèmes, depuis de purs problèmes de maximisation d'utilité jusqu'à des problèmes impliquant incertitudes, ob-

\* Cet article est la version française d'un article publié à ECAI06. Le travail décrit a par ailleurs été partiellement mené dans le cadre du projet européen COGNIRON (*The Cognitive Companion*).

servation partielle et infaisabilités. De nombreux formalismes ont été proposés pour les modéliser et les résoudre. Notre but est ici de définir un cadre général à base de modèle graphique permettant de capturer la localité de l'information et les indépendances, enrichi par une structure algébrique permettant de définir des algorithmes et d'établir des propriétés. Toutes les preuves sont omises par manque de place, mais sont disponibles à [26].

Par la suite,  $Dom(x)$  dénote l'ensemble des valeurs d'une variable  $x$ . Par extension, pour un ensemble de variables  $S$ ,  $Dom(S) = \prod_{x \in S} Dom(x)$ . Une fonction locale sur  $E$  est une fonction  $L : Dom(S) \rightarrow E$ , où  $S$  est un ensemble de variables. L'élimination d'un ensemble de variables  $S' \subseteq S$  de  $L$  avec un opérateur associatif et commutatif  $op$  sur  $E$  produit une fonction de  $S - S'$  dans  $E$  définie par  $(op_{S'} L)(A) = op_{A' \in Dom(S')} L(A.A')$  pour toute affectation  $A$  de  $S - S'$ . Les valeurs booléennes sont notées  $t$  et  $f$ .

Nous débutons avec un exemple basique impliquant uniquement des utilités.

**Exemple** Jean se trouve face à trois portes (portes  $A$ ,  $B$  et  $C$ ). L'une de ces portes cache un trésor. Une autre cache un gangster. Le gangster lui dérobera 4,000€. Le trésor vaut 10,000€.

**Modélisation** Pour représenter de façon compacte l'environnement et la décision, nous introduisons trois variables : (1) deux variables d'environnement : une pour la porte derrière laquelle se trouve le gangster (notée  $ga$ ) et une pour celle derrière laquelle se trouve le trésor (notée  $tr$ ) ; (2) une variable de décision (notée  $do$ ) représentant la porte que Jean décide d'ouvrir. Chaque variable a pour domaine  $\{A, B, C\}$ . Les variables de décision sont des variables dont la valeur est contrôlée par un agent, alors que les variables d'environnement sont des variables dont la valeur n'est pas directement contrôlée par un agent.

Pour représenter les utilités, nous avons besoin de deux fonctions locales d'utilité  $U_1$  et  $U_2$  : (1)  $U_1$  exprime que si Jean ouvre la porte du gangster, il devra payer 4,000€ (contrainte souple  $do = ga$ , avec un degré d'utilité de  $-4,000€$  si elle est satisfaite et de 0 sinon) ; (2)  $U_2$  exprime que si Jean ouvre la porte du trésor, il gagnera 10,000€ (contrainte souple  $do = tr$ , avec un degré d'utilité de 10,000€ si elle est satisfaite et de 0 sinon).

**Requête associée** Quelle porte Jean doit ouvrir s'il sait que le gangster se trouve derrière la porte  $A$  et le trésor derrière la porte  $C$  (absence d'incertitude) ? Évidemment, il doit ouvrir la porte  $C$ .

## 1.1 Ajouter des incertitudes

Dans des situations réelles, l'état de l'environnement peut ne pas être complètement connu : il peut exister des incertitudes sur cet environnement (ici dénommées *plausibilités*), mais aussi des observations de cet environnement incertain. Nous sophistiquons notre problème de recherche de trésor pour intégrer ces aspects.

**Exemple** Le trésor et le gangster ne sont pas derrière la même porte et toutes les situations respectant cette contrainte sont équiprobables.

Jean est accompagné de Pierre. Chacun d'entre eux peut décider d'écouter à l'une des portes pour essayer d'entendre le chien qui se trouve avec le gangster. La probabilité d'entendre le chien est de 0.8 si l'on écoute à la porte du gangster, de 0.4 si l'on écoute à une porte adjacente et de 0 sinon.

**Modélisation** Pour capturer cette nouvelle spécification du problème, nous introduisons (1) quatre variables supplémentaires : deux variables de décision  $li_J$  et  $li_P$ , avec  $\{A, B, C\}$  comme domaine, représentant les portes auxquelles Jean et Pierre décident d'écouter ; deux variables d'environnement  $he_J$  et  $he_P$ , avec  $\{yes, no\}$  comme domaine, représentant le fait que Jean et Pierre entendent ou non le chien du gangster ; (2) quatre fonctions locales de plausibilité :  $P_1 : ga \neq tr$  et  $P_2 = 1/6$  représentant la distribution de probabilité sur les positions respectives du trésor et du gangster ;  $P_3 = P(he_J | li_J, ga)$  représentant la probabilité que Jean entende le chien, étant données la porte à laquelle il écoute et celle derrière laquelle se trouve le gangster ; de façon similaire,  $P_4 = P(he_P | li_P, ga)$ . Implicitement, les plausibilités locales satisfont des conditions de normalisation. Tout d'abord, comme le trésor et le gangster sont quelque part,  $\sum_{ga, tr} P_1 \times P_2 = 1$ . Ensuite, comme Jean et Pierre entendent quelque chose ou non,  $\sum_{he_J} P_3 = 1$  et  $\sum_{he_P} P_4 = 1$ .

**Requêtes associées** Quelles sont les règles de décision qui maximisent l'utilité espérée, si Jean et Pierre commencent par écouter et si Jean décide ensuite d'ouvrir une porte sachant ce que l'un et l'autre ont entendu ?

Une approche classique pour répondre à une telle requête consiste à utiliser un arbre de décision. Dans cet arbre, les variables peuvent être considérées dans l'ordre  $li_J \rightarrow li_P \rightarrow he_J \rightarrow he_P \rightarrow do \rightarrow ga \rightarrow tr$  (d'abord, Jean et Pierre choisissent chacun une porte à laquelle écouter ; ensuite, ils entendent ou non le chien ; suivant ce qu'ils ont entendu, Jean décide quelle porte ouvrir ; finalement le gangster et le trésor sont chacun derrière une porte avec une certaine probabilité). Dans cet arbre, un nœud interne  $n$  correspond à une variable  $x$  et un arc de  $n$  vers un de ses nœuds fils est étiqueté par une affectation  $x = a$ . Si  $x$  est une variable d'environnement, cet arc est de plus pondéré par la probabilité  $P(x = a | A)$ , où  $A$  désigne l'affectation correspondant au chemin de la racine à  $x$ .

L'utilité d'un nœud feuille est l'utilité globale  $(U_1 + U_2)(A)$  de l'affectation complète  $A$  qui lui est associée. L'utilité d'un nœud interne de décision est donnée par la valeur d'un nœud fils optimal (et il est possible de mémoriser la décision optimale associée). L'utilité d'un nœud interne d'environnement est donnée par la somme des utilités de ses nœuds fils pondérées par les probabilités associées. L'utilité espérée globale est l'utilité du nœud racine. On peut montrer [26] que le résultat d'une telle procédure à base d'arbre de décision est identique au résultat du calcul

de la quantité suivante :

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

Des règles de décision optimale peuvent être mémorisées au cours de ce calcul. En d'autres termes, la procédure à base d'arbre de décision est équivalente à une séquence d'éliminations de variables sur une combinaison de fonctions locales.

Différentes séquences d'éliminations capturent différents problèmes ou situations: si Jean pense que Pierre est un traître et lui laisse choisir en premier la porte à laquelle écouter (attitude pessimiste concernant l'autre agent), la séquence d'éliminations  $\min_{li_P} \max_{li_J} \sum_{he_J, he_P} \max_{do} \sum_{ga, tr}$  est adéquate ( $li_P$  est éliminé avec min). Si l'on suppose que Pierre ne dit même pas à Jean ce qu'il a entendu (Jean n'observe pas  $he_P$ ), la séquence d'éliminations adéquate devient  $\min_{li_P} \max_{li_J} \sum_{he_J} \max_{do} \sum_{he_P, ga, tr}$ .

## 1.2 Ajouter des faisabilités

**Exemple** Jean et Pierre ne peuvent pas écouter à la même porte et la porte A est bloquée.

**Modélisation** Pour intégrer de tels aspects, nous ajoutons deux fonctions locales de faisabilité:  $F_1 : li_J \neq li_P$  et  $F_2 : do \neq A$ . Nous supposons que, dans toute situation, au moins une décision est faisable (pas de blocage). Ceci peut être représenté par des conditions de normalisation sur les faisabilités:  $\forall_{li_J, li_P} F_1 = t$  et  $\forall_{do} F_2 = t$ . Une procédure à base d'arbre de décision est alors équivalente au calcul de la quantité suivante :

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

qui utilise un opérateur spécial  $\star$  pour masquer les décisions infaisables: si  $\diamond$  désigne l'utilité de situations infaisables, alors  $\diamond$  doit être neutre pour les opérateurs d'élimination ( $op(e, \diamond) = e$  pour tout opérateur d'élimination  $op$ ) et absorbant pour les opérateurs de combinaison ( $\diamond \otimes e = \diamond$  pour tout opérateur de combinaison  $\otimes$ ) puisque la combinaison d'une décision infaisable avec n'importe quelle autre décision est elle-même infaisable. Ensuite,  $\star$  est l'opérateur qui, d'une part, associe à toute décision infaisable une utilité égale à  $\diamond$  et, d'autre part, ne modifie pas l'utilité d'une décision faisable:  $\star$  est défini par  $f \star \alpha = \diamond$  et  $t \star \alpha = \alpha$ . Ensemble,  $\diamond$  et  $\star$  permettent de réaliser des éliminations sur des domaines incluant des situations infaisables.

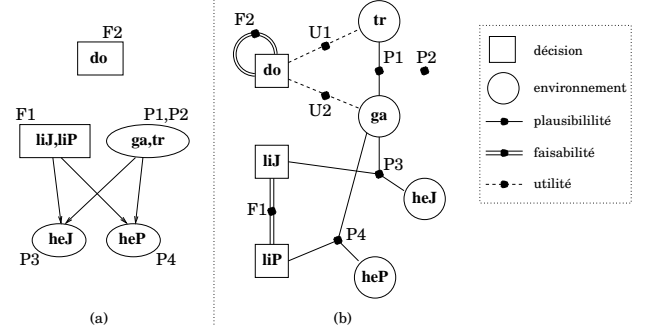


FIGURE 1 – Modèle graphique composite (a) DAG capturant les conditions de normalisation ; (b) Réseau de fonctions locales.

Globalement, la connaissance modélisée par des variables et des fonctions locales forme un *modèle graphique composite* défini par (a) un *graphe orienté acyclique* (DAG) qui capture les conditions de normalisation sur les plausibilités et les faisabilités (figure 1(a))<sup>1</sup> et (b) un *réseau de fonctions locales* qui inclut différents types de variables (variables de décision et d'environnement) et différents types de fonctions locales (fonctions locales de plausibilité, de faisabilité et d'utilité) (figure 1(b)).

## 1.3 D'autres opérateurs pour les plausibilités et les utilités

L'exemple précédent utilise des probabilités pour modéliser les incertitudes. Sous l'hypothèse d'indépendance, les probabilités sont combinées avec un opérateur  $\otimes_p = \times$  et éliminées (marginalisées) avec un opérateur  $\oplus_p = +$ . Cependant, d'autres théories de l'incertain, comme les possibilités [11] ou les  $\kappa$ -rankings [32], utilisent d'autres instantiations de  $\otimes_p$  et de  $\oplus_p$ .

Les utilités considérées jusqu'à présent sont additives (combinées avec  $\otimes_u = +$ ). Cependant, les utilités peuvent exprimer des préférences combinées avec  $\otimes_u = \min$ .

L'utilité espérée probabiliste est définie par  $\sum_i p_i \times u_i$ : plausibilités et utilités sont combinées avec  $\otimes_{pu} = \times$  et l'élimination des utilités pondérées par des plausibilités est réalisée avec  $\oplus_u = +$ . D'autres théories utilisent une utilité espérée définie par  $\oplus_u (p_i \otimes_{pu} u_i)$ , faisant appel à d'autres instantiations de  $\oplus_u$  et de  $\otimes_{pu}$ .

Le tableau 1 présente les ensembles et opérateurs utilisés pour représenter et manipuler plausibilités et utilités dans cinq cadres classiques.

**Vers un cadre générique** Après cette introduction informelle, il est possible de donner une définition formelle du cadre *Plausibilité-Faisabilité-Utilité* (PFU). Ce cadre combine dans un *modèle graphique algébrique* (Section 2)

1. Si  $P$  désigne l'ensemble des fonctions locales de plausibilité associées à un nœud du DAG correspondant à un ensemble de variables  $S$ , cela signifie que  $\sum_S (\prod_{P_i \in P} P_i) = 1$ . Si  $F$  désigne l'ensemble des fonctions locales de faisabilité associées à un nœud du DAG correspondant à un ensemble de variables  $S$ , cela signifie que  $\forall_S (\bigwedge_{F_i \in F} F_i) = t$ .

	$E_p$	$\oplus_p$	$\otimes_p$	$E_u$	$\otimes_u$	$\oplus_u$	$\otimes_{pu}$
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 – Ensembles et opérateurs avec (1) utilité espérée probabiliste, (2) satisfaction espérée probabiliste, utilité espérée possibiliste (3) optimiste et (4) pessimiste, (5) utilité qualitative avec  $\kappa$ -rankings et utilités positives.

des concepts des modèles graphiques (localité, indépendance conditionnelle) et une définition algébrique flexible de l'utilité espérée. Nous montrons ensuite comment des requêtes sur de tels modèles nous permettent d'exprimer et de résoudre des problèmes de décision.

Idéalement, de telles requêtes devraient pouvoir se ramener à une séquence d'éliminations de variables sur une combinaison de fonctions locales telle que :

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

## 2 Un modèle graphique algébrique

### 2.1 Une structure algébrique

**Structure de plausibilité** Nous définissons une *structure de plausibilité* comme un triplet  $(E_p, \oplus_p, \otimes_p)$ , où  $E_p$  est un ensemble de degrés de plausibilité équipé d'un ordre partiel  $\preceq_p$ ,  $\oplus_p$  est un opérateur d'élimination sur les plausibilités et  $\otimes_p$  un opérateur de combinaison sur les plausibilités.  $\oplus_p$  et  $\otimes_p$  satisfont un ensemble d'axiomes inspirés des mesures algébriques de plausibilité de Halpern [16]. La différence avec le travail d'Halpern est que nous étendons les opérateurs  $\oplus_p$  et  $\otimes_p$  de façon à ce qu'ils soient clos sur  $E_p$ . Ceci nous permet de définir  $(E_p, \oplus_p, \otimes_p)$  comme un semi-anneau commutatif (l'élément neutre de  $\oplus_p$  est noté  $0_p$  et l'élément neutre de  $\otimes_p$  est noté  $1_p$ ), où  $\oplus_p$  et  $\otimes_p$  sont monotones et où  $0_p$  (associé aux situations impossibles) est l'élément minimum de  $E_p$ .

**Structure d'utilité** Nous définissons une *structure d'utilité* comme une paire  $(E_u, \otimes_u)$ , où  $E_u$  est un ensemble de degrés d'utilité équipé d'un ordre partiel  $\preceq_u$  et où  $\otimes_u$  est un opérateur de combinaison sur les utilités. Nous supposons que  $(E_u, \otimes_u)$  est un semi-groupe commutatif ( $\otimes_u$  associatif et commutatif de façon à ce que les utilités combinées ne dépendent pas de la façon dont la combinaison est réalisée) et que  $\otimes_u$  est monotone.

**Structure d'utilité espérée** Pour prendre simultanément en compte plausibilités et utilités, nous définissons une *structure d'utilité espérée*. Cette structure inspirée, comme dans les MDP algébriques [25], des travaux de Chu et Halpern sur l'utilité espérée généralisée [8], est un quadruplet  $(E_p, E_u, \oplus_u, \otimes_{pu})$ , où  $\oplus_u$  (opérateur sur  $E_u$ ) et

$\otimes_{pu} : E_p \times E_u \rightarrow E_u$  définissent la formule de l'utilité espérée généralisée  $\oplus_u(p_i \otimes_{pu} u_i)$ . Nous étendons [8] de façon à pouvoir traiter des problèmes de décision *séquentielle*. Pour des raisons opérationnelles,  $\oplus_u$  et  $\otimes_{pu}$  sont aussi clos. Cela définit  $(E_p, E_u, \oplus_u, \otimes_{pu})$  comme un semi-module sur  $(E_p, \oplus_p, \otimes_p)$ , avec en plus des axiomes de monotonie de  $\oplus_u$  et  $\otimes_{pu}$ . On notera que toutes les structures du tableau 1 sont des structures d'utilité espérée.

**Hypothèses implicites** Comme un ensemble de degrés de plausibilité et un ensemble de degrés d'utilité sont définis, les plausibilités et les utilités doivent être *cardinales* : des approches purement ordinales, comme par exemple les CP-nets [4], ne sont pas couvertes. Comme  $\otimes_{pu}$  prend ses valeurs dans  $E_u$ , une hypothèse implicite est que plausibilités et utilités sont *commensurables* : des travaux tels que [13] ne sont donc pas couverts. Finalement, certains axiomes impliquent que seules des plausibilités *distributionnelles* puissent être considérées (la plausibilité d'un ensemble  $S$  d'affectations de variables est déterminée par les plausibilités de chacune des affectations de  $S$ ) : les fonctions de croyance [29] ne sont donc pas couvertes non plus.

**Structure algébrique pour le problème de recherche du trésor** La structure de plausibilité est  $(\mathbb{R}^+, +, \times)$  ( $E_p = \mathbb{R}^+$  et non  $[0, 1]$  de façon à ce que  $\oplus_p$  soit clos sur  $E_p$ ), la structure d'utilité est  $(\mathbb{R}, +)$  et la structure d'utilité espérée est  $(\mathbb{R}^+, \mathbb{R}, +, \times)$ .

### 2.2 Un modèle graphique

Suivant les concepts des modèles graphiques, l'environnement et les décisions sont représentés par des variables d'environnement et de décision, tandis que les relations entre ces variables (relations de plausibilité, de faisabilité et d'utilité) sont représentées par des fonctions locales. Ensemble, variables et fonctions locales définissent un modèle graphique composite et générique, appelé *réseau de plausibilité, de faisabilité et d'utilité* (ou *réseau PFU*).

**Définition 1** Un *réseau PFU* est un quintuplet  $(V, G, P, F, U)$  où :

- $V$  est un ensemble de variables, partitionné en  $V_D$  (ensemble des variables de décision) et  $V_E$  (ensemble des variables d'environnement).
- $G$  est un DAG dont les sommets forment une partition de  $V$  (les sommets sont des ensembles de variables appelés composants) tel que chaque sommet de  $G$  est

composé de variables de même nature (décision ou environnement). Nous notons  $\mathcal{C}_D$  l'ensemble des composants de décision et  $\mathcal{C}_E$  l'ensemble des composants d'environnement.

- $P = \{P_1, P_2, \dots\}$  est un ensemble fini de fonctions locales de plausibilité; chaque  $P_i$  est associée à un composant  $c \in \mathcal{C}_E$ , noté  $c(P_i)$ , tel que  $S(P_i) \subseteq (c \cup pa_G(c))$ . Pour chaque composant  $c \in \mathcal{C}_E$ ,  $\bigoplus_{P_i: c(P_i)=c} P_i = 1_p$ .
- $F = \{F_1, F_2, \dots\}$  est un ensemble fini de fonctions locales de faisabilité; chaque  $F_i$  est associée à un composant  $c \in \mathcal{C}_D$ , noté  $c(F_i)$ , tel que  $S(F_i) \subseteq (c \cup pa_G(c))$ . Pour chaque composant  $c \in \mathcal{C}_D$ ,  $\bigvee_{F_i: c(F_i)=c} F_i = t$ .
- $U = \{U_1, U_2, \dots\}$  est un ensemble fini de fonctions locales d'utilité.

La décomposition des degrés globaux de plausibilité, de faisabilité et d'utilité en un ensemble de fonctions locales est sémantiquement justifiée par la notion d'*indépendance conditionnelle*. Nous en proposons une justification intuitive en ce qui concerne les plausibilités. Disons que  $\mathcal{P}_S$  est une distribution de plausibilité sur  $S$  si et seulement si  $\bigoplus_{P \in \mathcal{P}_S} P = 1_p$  (dans le cas des probabilités, cela signifie simplement que la somme des probabilités est égale à 1).  $\mathcal{P}_S$  définit une distribution de plausibilité sur tout sous-ensemble  $S'$  de  $S$ :  $\mathcal{P}_{S'} = \bigoplus_{P \in \mathcal{P}_S} P|_{S'}$ .

Pour définir l'indépendance conditionnelle, nous définissons une fonction de conditionnement  $\odot_p$  (vérifiant certaines propriétés) qui nous permet de définir des distributions de plausibilité conditionnelles:  $\mathcal{P}_{S_1|S_2} = \mathcal{P}_{S_1, S_2} \odot_p \mathcal{P}_{S_2}$  (dans le cas des probabilités,  $\odot_p$  est la division). Ensuite, pour tout triplet d'ensembles de variables disjoints  $S_1, S_2$  et  $S_3$ , nous disons que  $S_1$  est *conditionnellement indépendant* de  $S_2$  étant donné  $S_3$  si et seulement si  $\mathcal{P}_{S_1, S_2|S_3} = \mathcal{P}_{S_1|S_3} \odot_p \mathcal{P}_{S_2|S_3}$ .

Finalement, un DAG  $G$  de composants est dit *compatible* avec une distribution de plausibilité  $\mathcal{P}_S$  si et seulement si, pour tout composant  $c$  de  $G$ ,  $c$  est conditionnellement indépendant de ses non descendants dans  $G$ , étant donné l'ensemble  $pa_G(c)$  de ses parents dans  $G$ . De la même façon que pour les réseaux bayésiens [24], on peut démontrer [26] le théorème suivant :

**Theorem 1** *Si  $G$  est un DAG compatible avec  $\mathcal{P}_S$ , alors  $\mathcal{P}_S = \bigotimes_{P \in \mathcal{P}_S} P|_{pa_G(c)}$ .*

Grâce au théorème 1<sup>2</sup>, un DAG compatible avec une distribution de plausibilité globale permet de la factoriser comme une combinaison de fonctions locales  $L_{c, pa_G(c)}$  ( $L_{c, pa_G(c)} = \mathcal{P}_c|_{pa_G(c)}$ ) telles que  $\bigoplus_{P \in \mathcal{P}_S} L_{c, pa_G(c)} = 1_p$  pour tout composant  $c$  de  $G$ . Chaque  $L_{c, pa_G(c)}$  peut être elle-même factorisée comme une combinaison de fonctions

2. Et avec quelques étapes dues au fait que nous ne travaillons pas sur des distributions de plausibilité sur  $V$ , mais sur des distributions de plausibilité sur  $V_E$  pour toute affectation de  $V_D$ .

locales de plausibilité  $P_i$  associées à  $c$  (association notée  $c(P_i) = c$ ) vérifiant donc  $\bigoplus_{P_i: c(P_i)=c} P_i = 1_p$ . Cette double factorisation explique l'origine du DAG dans un réseau PFU et les conditions de normalisation qu'il encode.

Des résultats similaires peuvent être établis pour les faisabilités. Quant aux utilités, les réseaux PFU font l'hypothèse implicite qu'une distribution d'utilité globale  $\mathcal{U}_V$  sur l'ensemble des variables peut être factorisée comme une combinaison de fonctions locales d'utilité  $U_i$ . On peut supposer que cette factorisation est obtenue directement, comme cela se pratique avec les CSP [21] ou qu'elle est justifiée par une notion d'indépendance conditionnelle dans le cas  $\otimes_u = +$ , comme dans [1].

**Exemple** Le réseau PFU de notre exemple apparaît dans la figure 1.

**Formalismes couverts** Un CSP [21] peut être représenté comme un réseau PFU  $(V, G, \emptyset, \emptyset, U)$ , où  $V$  est l'ensemble des variables du CSP,  $G$  est un DAG avec  $V$  comme unique composant de décision et  $U$  est l'ensemble des contraintes du CSP (si les contraintes étaient modélisées comme des faisabilités, les conditions de normalisation ne seraient pas satisfaites pour des CSP incohérents). Il est aussi possible de spécifier les réseaux PFU associés à des CSP *valués* [2], *quantifiés* [3], *mixtes* [12] ou *stochastiques* [33], ainsi qu'à des problèmes SAT, QBF (*Quantified Boolean Formulas*) ou (*Extended*) *Stochastic SAT* [20].

Un *réseau bayésien* [24] peut être représenté comme un réseau PFU  $(V, G, P, \emptyset, \emptyset)$ , où  $G$  est le DAG associé au réseau bayésien et  $P$  est l'ensemble des distributions de probabilité locales associées au réseau. *Graphes chaînés* [14] et *champs aléatoires markoviens* [7] sont aussi couverts.

Un MDP *probabiliste* à horizon fini [27, 22] peut être représenté par un réseau PFU  $(V, G, P, \emptyset, U)$ , où  $V_D$  est l'ensemble des décisions  $d_t$  (une par instant),  $V_E$  est l'ensemble des états  $s_t$  (un par instant),  $G$  est un DAG représentant le MDP déroulé,  $P$  est l'ensemble des distributions locales de probabilité  $P_{s_{t+1}|s_t, d_t}$  et  $U$  l'ensemble des gains locaux additifs  $R_{s_t, d_t}$ . Il est aussi possible de représenter des MDP *possibilistes* à horizon fini [28], des MDP à base de  $\kappa$ -rankings et des MDP *partiellement observables* et *factorisés* [6, 5].

Un *diagramme d'influence* [17], avec une possible variante possibiliste, peut être représenté par un réseau PFU  $(V, G, P, \emptyset, U)$ . Avec les *réseaux de valuation* [31], un ensemble  $F$  de fonctions locales de faisabilité doit être ajouté.

### 3 Raisonner sur des réseaux PFU via des requêtes

Dans cette section, nous supposons qu'une séquence de décisions doit être réalisée et que l'ordre selon lequel les décisions sont prises et l'environnement est observé est connu. Nous faisons aussi une hypothèse de *non oubli*, ce qui signifie que, quand un agent prend une décision, il connaît toutes les décisions et observations précédentes. Finalement, l'ordre sur les degrés d'utilité est supposé total.

Sous ces hypothèses, nous voulons exprimer des problèmes de décision séquentielle sur des réseaux PFU, prenant en compte les possibilités d'observation partielle et de présence d'agents coopératifs ou non. Pour représenter de tels problèmes, nous utilisons une séquence *Sov* de paires opérateur-variable(s) qui capture :

- de possibles inobservabilités : l'ordre selon lequel les décisions sont prises et les observations sont faites est spécifié par *Sov*. Par exemple, si la valeur de  $he_P$  est connue quand Jean choisit la porte à ouvrir, *Sov* contient  $\dots (\oplus_u, he_P) \dots (\max, do) \dots$ . Sinon, une séquence telle que  $\dots (\max, do) \dots (\oplus_u, he_P) \dots$  est utilisée ;
- une attitude optimiste ou pessimiste vis-à-vis des autres agents décideurs. Par exemple, si Jean pense que Pierre agira au mieux (il sait que Pierre est coopératif ou il est simplement optimiste au sujet de sa décision), *Sov* contient  $(\max, li_P)$ . Dans le cas contraire,  $(\min, li_P)$  est utilisé.

**Exemple** Pour la dernière requête de la section 1.1, *Sov* est égale à  $(\min, \{li_P\}).(\max, \{li_J\}).(+, \{he_J\}).(\max, \{do\}).(+, \{he_P, ga, tr\})$ .

**Definition 2** Une requête *Q* sur un réseau PFU est une paire  $(\mathcal{N}, Sov)$  où (1)  $\mathcal{N}$  est un réseau PFU ; (2) *Sov* est une séquence de paires opérateur-variable(s) telle que les opérateurs soient  $\min$ ,  $\max$  ou  $\oplus_u$  et que chaque variable apparaisse au plus une fois dans *Sov*.

**Requêtes correctes** Certaines requêtes n'ont pas de sens. La principale condition pour qu'une requête soit correcte est qu'elle ne contienne pas une paire  $x, y$  de variables de natures différentes (l'une d'environnement, l'autre de décision) telles que le composant de  $x$  soit ascendant du composant de  $y$  dans le DAG et que  $x$  apparaisse après  $y$  dans *Sov*. Informellement, cela signifierait que  $y$  est affecté avant  $x$  en contradiction avec la *causalité*. Par exemple, la séquence suivante  $\dots (+, he_J) \dots (\max, li_J) \dots$  n'est pas correcte puisque Jean ne peut pas entendre quelque chose avant d'avoir choisi la porte à laquelle écouter.

### 3.1 Répondre à des requêtes

**Une approche à base d'arbre de décision** Une première approche pour répondre à des requêtes dans le cas général (pas seulement dans le cas de l'utilité espérée probabiliste) utilise des arbres de décision. Les variables sont considérées dans l'ordre de la séquence *Sov*. Les arcs issus des nœuds internes d'environnement sont pondérés par des plausibilités conditionnelles de la forme  $\mathcal{P}(x = a | A)$  et les arcs issus des nœuds internes de décision sont pondérés par des faisabilités conditionnelles de la forme  $\mathcal{F}(x = a | A)$ .

L'utilité espérée d'une requête et les règles de décision associées peuvent alors être obtenues par une procédure similaire à celle décrite dans la section 1.1 : l'utilité d'un nœud feuille est donnée par la combinaison des utilités locales

$U_i$ , l'utilité d'un nœud interne d'environnement est donnée par la somme pondérée des utilités de ses nœuds fils et l'utilité d'un nœud interne de décision est donnée par l'utilité optimale parmi ses fils faisables.

**Une approche plus opérationnelle** L'avantage de l'approche à base d'arbre de décision est qu'elle a des justifications sémantiques claires. Au delà de la taille potentiellement exponentielle de l'arbre de décision, son défaut est cependant que chaque nœud interne dans l'arbre peut nécessiter le calcul de  $\mathcal{P}(x = a | A)$  ou  $\mathcal{F}(x = a | A)$ , aussi potentiellement exponentiel en temps. Il est heureusement possible de montrer [26] que l'approche à base d'arbre de décision (appelée *réponse sémantique* à *Q*) est équivalente à une approche algébrique directe (appelée *réponse opérationnelle* à *Q*) qui n'a elle besoin que des fonctions locales disponibles dans le réseau PFU initial.

**Theorem 2** Répondre à une requête via un arbre de décision équivaut à calculer :

$$Ans(Q) = Sov \left( \left( \bigwedge_{F_i \in F} F_i \right) \star \left( \bigotimes_{P_i \in P} P_i \right) \otimes_{pu} \left( \bigotimes_{U_i \in U} U_i \right) \right)$$

De plus, les règles de décision optimale obtenues au cours de ce calcul sont les mêmes que celles obtenues via une approche à base d'arbre de décision.

### 3.2 Requêtes classiques couvertes

La plupart des requêtes usuelles dans les formalismes existants peuvent se ramener à des requêtes sur des réseaux PFU : trouver une solution à un problème SAT, à un CSP [21] ou à un CSP *valué* [2] correspond à une séquence  $Sov = (\max, V)$ . Pour des QBF ou des CSP *quantifiés* [3], *Sov* alterne des éliminations avec  $\min$  (en cas de quantificateur universel) et avec  $\max$  (en cas de quantificateur existentiel). Avec les CSP *mixtes* ou *probabilistes* [12], *Sov* est de la forme  $(\oplus_u, V_E).(\max, V_D)$  en cas de recherche d'une décision conditionnelle et de la forme  $(\max, V_D).(\oplus_u, V_E)$  en cas de recherche d'une décision inconditionnelle. La situation est similaire dans un contexte de *planification* dite *conformante* ou *probabiliste* [15].

Une requête sur un *réseau bayésien* [24] est de la forme  $(+, S)$  pour le calcul d'une distribution de probabilité sur  $V - S$ , de la forme  $(\max, V)$  pour résoudre un problème d'explication la plus probable (MPE) et de la forme  $(\max, V_D).(+, V_E)$  pour résoudre un problème d'hypothèse la plus vraisemblable (MAP).

Avec des CSP *stochastiques* [33] ou des *diagrammes d'influence* [17], la séquence alterne  $+$  sur les variables d'environnement et  $\max$  sur les variables de décision. Avec les MDP à horizon fini, *Sov* est de la forme  $(\max, d_1).(\oplus_u, s_2) \dots (\max, d_T).(\oplus_u, s_T)$ . Avec des POMDP à horizon fini, on ajoute des observations  $o_t$  (une par instant) et  $Sov = (\max, d_1).(\oplus_u, o_2) \dots (\max, d_T).(\oplus_u, o_T).(\oplus_u, \{s_1, \dots, s_T\})$  (l'état reste non observé).

## 4 Gains et coûts

**Une meilleure compréhension** Dans la mesure où il couvre de nombreuses requêtes sur des modèles graphiques existants, le cadre PFU permet une compréhension plus précise des similarités et des différences entre ces différents formalismes. Il définit une base commune permettant une communication entre les différentes communautés.

**Une expressivité accrue** Le cadre PFU permet une grande variabilité : (1) variabilité au niveau de la structure algébrique utilisée (utilité espérée probabiliste, satisfaction espérée probabiliste, utilité pessimiste possibiliste, utilité optimiste possibiliste ou utilité qualitative avec  $\kappa$ -rankings); (2) variabilité au niveau du réseau qui exploite des indépendances orientées ou non, ainsi que des conditions de normalisation; (3) variabilité au niveau des requêtes qui permettent de modéliser différentes situations en termes d'observabilité partielle ou non et d'attitude optimiste ou pessimiste vis-à-vis des autres agents.

Le cadre PFU est donc plus expressif que chacun des formalismes qu'il couvre et il inclut des formalismes non encore publiés, tels que par exemple les *diagrammes d'influence possibilistes* ou encore les *CSP stochastiques étendus* pour prendre en compte le fait que les décisions peuvent influencer sur l'environnement.

**Des algorithmes génériques** Calculer une réponse à une requête correcte est évidemment *PSPACE-dur*. Il est par ailleurs facile de définir, à partir du théorème 2, un algorithme de recherche arborescente qui calcule la réponse à une requête correcte et qui soit polynomial en espace. De façon similaire, il est possible de définir un algorithme à base d'élimination de variables pour calculer  $Ans(Q)$  [10]. Le cadre algébrique PFU est une opportunité pour identifier des conditions nécessaires et suffisantes à l'application d'algorithmes existants [23] ou pour définir de nouvelles techniques dont chaque formalisme couvert pourrait bénéficier. Des techniques de type calcul de bornes ou cohérence locale [21, 9, 19] pourraient par exemple être intégrées pour accélérer la résolution.

Finalement, le cadre PFU peut être vu comme une opportunité pour intégrer dans un cadre générique des techniques développées dans les différents formalismes couverts et permettre ainsi une fertilisation réciproque.

## 5 Conclusion

Dans ce papier, un cadre algébrique générique pour la décision séquentielle a été introduit. Il combine une structure algébrique qui spécifie comment combiner et synthétiser l'information avec un modèle graphique qui spécifie des fonctions locales de plausibilité, de faisabilité et d'utilité. Des requêtes permettent ensuite de modéliser diverses situations en termes d'observabilité partielle ou non ou d'attitude optimiste ou pessimiste vis-à-vis des autres agents décideurs.

Il est possible de répondre à ces requêtes en calculant une séquence d'éliminations de variables sur une combinai-

son de fonctions locales et la réponse fournie a une sémantique claire grâce à l'équivalence avec une approche à base d'arbre de décision. Comparé aux *algèbres de valuation* [30, 18] (un cadre générique pour réaliser des éliminations de variables sur une combinaison de fonctions locales), le cadre PFU utilise différents opérateurs de combinaison ( $\wedge, \star, \otimes_p, \otimes_{pu}, \otimes_u$ ) et d'élimination ( $\min, \max, \oplus_u$ ). De plus, les justifications sémantiques de la définition d'un réseau PFU, permettent d'inclure un DAG qui capture des conditions de normalisation.

Le cadre obtenu couvre non seulement de nombreuses requêtes sur des formalismes existants. Il permet aussi de définir des formalismes non encore publiés. D'un point de vue algorithmique, des schémas généraux qui intègrent les techniques utilisées dans les différents formalismes couverts peuvent être développés.

## Références

- [1] F. Bacchus and A. Grove. Graphical Models for Preference and Utility. In *Proc. of the 11th International Conference on Uncertainty in Artificial Intelligence (UAI-95)*, Montréal, Canada, 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. Montfroy. Beyond NP: Arc-consistency for Quantified Constraints. In *Proc. of the 8th International Conference on Principles and Practice of Constraint Programming (CP-02)*, pages 371–386, 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 International Conference on Uncertainty in Artificial Intelligence (UAI-99)*, Stockholm, Sweden, 1999.
- [5] C. Boutilier, T. Dean, and S. Hanks. Decision-Theoretic Planning: Structural Assumptions and Computational Leverage. *Journal of Artificial Intelligence Research*, 11:1–94, 1999.
- [6] C. Boutilier, R. Dearden, and M. Goldszmidt. Stochastic Dynamic Programming with Factored Representations. *Artificial Intelligence*, 121(1-2):49–107, 2000.
- [7] R. Chellappa and A. Jain, editors. *Markov Random Fields : Theory and Applications*. Academic Press, 1993.
- [8] F. Chu and J. Halpern. Great Expectations. Part I : On the Customizability of Generalized Expected Utility. In *Proc. of the 18th International Joint Conference on Artificial Intelligence (IJCAI-03)*, 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*, 113:41–85, 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 National Conference on Artificial Intelligence (AAAI-96)*, pages 175–180, Portland, OR, USA, 1996.
- [13] H. Fargier and P. Perny. Qualitative Models for Decision under Uncertainty without the Commensurability Assumption. In *Proc. of the 15th International Conference on Uncertainty in Artificial Intelligence (UAI-99)*, Stockholm, Sweden, 1999.
- [14] M. Frydenberg. The Chain Graph Markov Property. *Scandinavian Journal of Statistics*, 17:333–353, 1990.
- [15] M. Ghallab, D. Nau, and P. Traverso. *Automated Planning: Theory and Practice*. Morgan Kaufmann, 2004.
- [16] J. Halpern. Conditional Plausibility Measures and Bayesian Networks. *Journal of Artificial Intelligence Research*, 14:359–389, 2001.
- [17] R. Howard and J. Matheson. Influence Diagrams. In *Readings on the Principles and Applications of Decision Analysis*, pages 721–762. Strategic Decisions Group, Menlo Park, CA, USA, 1984.
- [18] J. Kohlas. *Information Algebras: Generic Structures for Inference*. Springer, 2003.
- [19] J. Larrosa and T. Schiex. In the Quest of the Best Form of Local Consistency for Weighted CSP. In *Proc. of the 18th International Joint Conference on Artificial Intelligence (IJCAI-03)*, pages 239–244, Acapulco, Mexico, 2003.
- [20] M. Littman, S. Majercik, and T. Pitassi. Stochastic Boolean Satisfiability. *Journal of Automated Reasoning*, 27(3):251–296, 2001.
- [21] A. Mackworth. Consistency in Networks of Relations. *Artificial Intelligence*, 8(1):99–118, 1977.
- [22] G. Monahan. A Survey of Partially Observable Markov Decision Processes: Theory, Models, and Algorithms. *Management Science*, 28(1):1–16, 1982.
- [23] P. Ndilikilikisha. Potential Influence Diagrams. *Journal of Approximated Reasoning*, 10:251–285, 1994.
- [24] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988.
- [25] P. Perny, O. Spanjaard, and P. Weng. Algebraic Markov Decision Processes. In *Proc. of the 19th International Joint Conference on Artificial Intelligence (IJCAI-05)*, pages 1372–1377, Edinburgh, Scotland, 2005.
- [26] C. Pralet, G. Verfaillie, and T. Schiex. An Algebraic Graphical Model for Decision with Uncertainties, Feasibilities, and Utilities. Technical report, LAAS-CNRS, 2005. <http://www.laas.fr/~cpralet/praletverfschiex.ps>.
- [27] M. Puterman. *Markov Decision Processes, Discrete Stochastic Dynamic Programming*. John Wiley & Sons, 1994.
- [28] R. Sabbadin. A Possibilistic Model for Qualitative Sequential Decision Problems under Uncertainty in Partially Observable Environments. In *Proc. of the 15th International Conference on Uncertainty in Artificial Intelligence (UAI-99)*, Stockholm, Sweden, 1999.
- [29] G. Shafer. *A Mathematical Theory of Evidence*. Princeton University Press, 1976.
- [30] P. Shenoy. Valuation-based Systems for Discrete Optimization. In *Proc. of the 6th International Conference on Uncertainty in Artificial Intelligence (UAI-90)*, pages 385–400, Cambridge, MA, USA, 1990.
- [31] P. Shenoy. Valuation Network Representation and Solution of Asymmetric Decision Problems. *European Journal of Operational Research*, 121:579–608, 2000.
- [32] W. Spohn. A General Non-probabilistic Theory of Inductive Reasoning. In *Proc. of the 6th International Conference on Uncertainty in Artificial Intelligence (UAI-90)*, Cambridge, MA, USA, 1990.
- [33] T. Walsh. Stochastic Constraint Programming. In *Proc. of the 15th European Conference on Artificial Intelligence (ECAI-02)*, Lyon, France, 2002.