

---

# Un cadre graphique et algébrique pour les problèmes de décision incluant incertitudes, faisabilités et utilités

Cédric Pralet\* \*\*\* — Gérard Verfaillie\*\* — Thomas Schiex\*\*\*

\* LAAS-CNRS  
7, av. du Colonel Roche, 31077 Toulouse Cedex 4  
cpralet@laas.fr

\*\* ONERA  
2, av. Édouard Belin, BP 4025, 31055 Toulouse Cedex 4  
gerard.verfaillie@onera.fr

\*\*\* INRA  
Chemin de Borde-Rouge, BP 52627, 31326 Castanet-Tolosan Cedex  
tschiex@toulouse.inra.fr

---

*RÉSUMÉ.* Dans cet article, nous présentons un cadre générique, graphique et algébrique, permettant de représenter et de manipuler des problèmes de décision séquentielle incluant incertitudes, faisabilités et utilités. Ce cadre, dénommé PFU pour Plausibilités, Faisabilités et Utilités, généralise de nombreux cadres proposés dans le domaine de l'Intelligence Artificielle : problèmes de satisfaisabilité, réseaux de contraintes, réseaux bayésiens, diagrammes d'influence, processus décisionnels markoviens, planification d'actions . . .

*ABSTRACT.* In this article, we present a generic graphical algebraic framework, which allows sequential decision-making problems including uncertainties, feasibilities, and utilities to be represented and handled. This framework, referred to as PFU for Plausibilities, Feasibilities, and Utilities, generalizes numerous frameworks that were proposed in the Artificial Intelligence community : satisfiability problems, constraint networks, Bayesian networks, influence diagrams, Markov decision processes, action planning . . .

*MOTS-CLÉS :* Modèles graphiques, Incertitudes, Faisabilités, Utilités

*KEYWORDS:* Graphical models, Uncertainties, Feasibilities, Utilities

---

## 1. Motivation

Divers cadres ont été proposés ou utilisés depuis plusieurs décennies par les chercheurs en Intelligence Artificielle pour modéliser et résoudre de façon automatique des problèmes de *décision*, éventuellement *séquentielle*, incluant *contraintes*, *incertitudes* et *préférences*. On peut entre autres citer :

- le cadre des problèmes de satisfiabilité d'une formule logique propositionnelle (*SAT*), avec ses extensions permettant de prendre en compte un contexte non déterministe (*Quantified Boolean Formulae, QBF*) ou stochastique (*Stochastic Satisfiability* (Littman *et al.*, 2001)) ;

- le cadre très proche des problèmes de satisfaction de contraintes ou réseaux de contraintes (*Constraint Satisfaction Problems, CSP* (Mackworth, 1977)), avec ses extensions permettant de prendre en compte des préférences (*Valued* ou *Semiring-based CSP* (Bistarelli *et al.*, 1999)) ou un contexte non déterministe (*Quantified CSP* (Bordeaux *et al.*, 2002), *Mixed CSP* (Fargier *et al.*, 1996)) ou stochastique (*Stochastic CSP* (Fargier *et al.*, 1995; Walsh, 2002)) ;

- le cadre des réseaux bayésiens (*Bayesian Networks* (Pearl, 1988)), avec ses extensions permettant de prendre en compte des décisions, des utilités et des faisabilités (*Influence Diagrams* (Howard *et al.*, 1984), *Valuation networks* (Shenoy, 1992)) ou un contexte séquentiel (*Dynamic Bayesian Networks* (Murphy, 2002)) ;

- le cadre des processus décisionnels markoviens (*Markov Decision Processes, MDP* (Pearl, 1988)), avec ses extensions permettant de prendre en compte un contexte d'observabilité partielle (*Partially Observable MDP* (Monahan, 1982)) ou la structure des états (*Factored MDP* (Boutilier *et al.*, 2000)) ;

- le cadre enfin de la planification (*STRIPS Planning* (Fikes *et al.*, 1971), *PDDL* (McDermott, 2000)) et ses extensions permettant de prendre en compte l'incertitude sur l'état courant et sur les effets des actions (*Conformant Planning* (Smith *et al.*, 1998), *Probabilistic Planning* (Kushmerick *et al.*, 1995)).

Au delà de leurs nombreuses différences, ces cadres possèdent d'importantes similitudes :

- ils mettent en jeu des *variables* à domaine souvent fini, qui peuvent représenter, soit l'état de l'*environnement*, soit les *décisions* de l'agent ou d'autres agents ;

- ils mettent en jeu des *relations* ou *fonctions locales* qui peuvent représenter, soit des *faisabilités* pesant sur les variables de décision (par exemple, des pré-conditions d'actions), soit des *incertitudes* ou *plausibilités* portant sur les variables d'environnement (par exemple, des distributions de probabilité conditionnelles), soit des *préférences* ou *utilités* fonctions des diverses variables (par exemple, des coûts ou des gains) ;

- ils font appel à divers opérateurs, soit pour *agrèger* les fonctions locales (par exemple, le  $\wedge$  logique pour agréger les faisabilités, le  $\times$  pour agréger les probabilités, le  $+$  pour agréger les utilités), soit pour *synthétiser* une information globale (par exemple, le  $\vee$  logique pour décider d'une faisabilité, le  $+$  pour calculer une probabilité

marginale, le max ou le min pour sélectionner une décision optimale).

Ils peuvent donc tous être vus comme des *modèles graphiques* dans la mesure où ils reposent tous, implicitement ou explicitement, sur un *hyper-graphe* de relations ou fonctions locales entre variables à domaine fini. Les différences entre eux tiennent essentiellement à ce que représentent variables et relations, ainsi qu'aux opérateurs d'agrégation et de synthèse utilisés.

D'où l'idée de les rassembler dans un même cadre *générique*, graphique et algébrique : *graphique* pour respecter leur nature graphique et *algébrique* pour abstraire les opérateurs d'agrégation et de synthèse utilisés et ne plus considérer que des opérateurs abstraits dotés de certaines propriétés algébriques. L'intérêt potentiel d'une telle unification est multiple :

- un cadre générique peut permettre de mieux comprendre des relations souvent ignorées entre des cadres spécifiques développés par des communautés qui souvent se méconnaissent ;

- il devrait permettre de développer une algorithmique générique dont on sait qu'elle se révèle souvent aussi performante, sinon plus performante, que les algorithmiques spécifiques développées dans tel ou tel cadre au prix d'efforts non négligeables ;

- il devrait aussi favoriser le dialogue entre communautés et permettre de récupérer dans une algorithmique commune les nombreuses idées développées et expérimentées dans ces communautés ;

- il peut enfin permettre, par la variété de la structure proposée, de considérer de nouveaux cadres spécifiques non encore explorés.

La variété des cadres visés, des problèmes de satisfiabilité aux processus décisionnels markoviens, peut cependant faire penser que leur unification est hors d'atteinte ou que le résultat en serait un monstre incompréhensible et ingérable. L'objectif de cet article est de montrer qu'il n'en est rien et que tous ces cadres sont de fait suffisamment proches pour être, grâce à une approche algébrique, réunis dans un seul, que nous dénommons PFU et dont les composants et les propriétés peuvent être décrits de façon relativement simple et compacte.

Dans son esprit, cette démarche d'unification est proche de celles qui ont conduit à des cadres graphiques et algébriques tels que les *Valuation algebras* (Shenoy, 1991; Kohlas, 2003), les *Valued* ou *Semiring-based CSP* (Bistarelli *et al.*, 1999) ou encore les *Algebraic MDP* (Perny *et al.*, 2005). Elle est simplement un peu plus ambitieuse en termes de cadres couverts.

Dans la section 2, nous introduisons informellement tous les ingrédients du cadre PFU via un exemple jouet que nous utilisons tout au long de l'article. Dans la section 3, nous présentons formellement le cadre PFU et ses trois composants : une structure algébrique générique, une structure graphique générique et une structure générique des requêtes. Nous présentons aussi ses propriétés basiques et les cadres spécifiques qu'il englobe. Dans la section 4, nous présentons un panorama préliminaire des algorithmes

génériques qu'il est possible de développer pour répondre de façon automatique à une requête. Dans la section 5, nous concluons sur le résultat courant et le chemin qui reste à parcourir.

Les limites en termes de nombre de pages nous interdisent d'inclure dans cet article les preuves des nombreux théorèmes qui jalonnent la démarche. Le lecteur intéressé pourra se reporter à un rapport technique accessible via Internet (Pralet *et al.*, 2005).

## 2. Un exemple

### 2.1. Une première version avec seulement des utilités

Considérons le problème suivant : *Jean a trois portes en face de lui : A, B et C de gauche à droite. Derrière l'une de ces portes, se trouve un trésor et derrière une autre, un gangster. Jean doit décider quelle porte ouvrir. Il sait qu'il gagnera 10000€ s'il ouvre la porte où se trouve le trésor, mais qu'il devra payer 4000€ s'il ouvre la porte où se trouve le gangster. On l'informe que le trésor est derrière la porte A et que le gangster est derrière la porte C.*

Pour modéliser ce problème trivial, nous introduisons trois variables : deux variables représentant l'environnement de Jean, une notée *tr* pour représenter la porte du trésor et l'autre notée *ga* pour représenter la porte du gangster, plus une variable notée *ou* représentant la décision de Jean. Ces trois variables ont toutes le même domaine de valeur  $\{A, B, C\}$ .

Pour modéliser les coûts et les gains possibles, nous introduisons deux fonctions locales d'utilité : une première  $U_1$  qui exprime que si Jean ouvre la porte du trésor, il gagne 10000€ (contrainte souple  $ou = tr$  de poids 10000, qui renvoie son poids si elle est satisfaite et 0 sinon) et une seconde  $U_2$  qui exprime que si Jean ouvre la porte du gangster, il paye 4000€ (contrainte souple  $ou = ga$  de poids  $-4000$ , qui renvoie de même son poids si elle est satisfaite et 0 sinon).

Dans cette situation sans incertitudes, une requête classique est : *Quelle est ou quelles sont la(les) décision(s) de Jean qui maximise(nt) son utilité ?* La requête formelle associée est de la forme :

$$\max_{ou} \left( \sum_{i=1}^2 U_i((tr = A), (ga = C)) \right) \quad [1]$$

En supposant qu'en calculant l'utilité maximale (le *max*), on mémorise la(les) valeur(s) qui la maximise(nt) (l'*argmax*), la réponse est évidemment  $ou = A$ .

Cet exemple met en évidence une distinction importante entre variables de *décision* contrôlées par un agent et variables d'*environnement* non contrôlées par un agent. Il montre aussi que des fonctions locales d'*utilité* peuvent s'exprimer sous la forme de

contraintes souples. Il montre enfin une *agrégation* classique des utilités via un  $+$  et une *synthèse* toute aussi classique du résultat de l'agrégation via un  $\max$ .

## 2.2. Une seconde version impliquant des plausibilités

Supposons maintenant que Jean sache seulement que le trésor et le gangster ne se trouvent pas derrière la même porte et que le trésor ne se trouve pas derrière la porte  $C$ .

L'incertitude sur la position du trésor et du gangster peut être modélisée par trois fonctions locales de plausibilité : une première  $P_1$  qui exprime que le trésor et le gangster ne sont pas derrière la même porte (contrainte  $tr \neq ga$ , qui renvoie 1 si elle est satisfaite et 0 sinon), une seconde  $P_2$  qui exprime que le trésor ne se trouve pas derrière la porte  $C$  (contrainte  $tr \neq C$  qui renvoie de même 1 si elle est satisfaite et 0 sinon) et une troisième  $P_3$  qui assure la normalisation de la distribution de probabilité sur  $tr$  et  $ga$  (constante  $P_3 = 1/4$  qui garantit que  $\sum_{tr,ga} P_1 \times P_2 \times P_3 = 1$ ).

Dans cette situation avec incertitudes, une requête classique est : *Quelle est ou quelles sont la(les) décision(s) de Jean qui maximise(nt) son utilité espérée ?* La requête formelle associée est de la forme :

$$\max_{ou} \sum_{tr,ga} \left( \left( \prod_{i=1}^3 P_i \right) \times \left( \sum_{i=1}^2 U_i \right) \right) \quad [2]$$

Cet exemple montre que des fonctions locales de *plausibilité* (ici des probabilités) peuvent s'exprimer sous la forme de contraintes, mais qu'une *normalisation* est nécessaire pour obtenir une distribution de plausibilité. Il montre aussi une *agrégation* classique des probabilités via un  $\times$  et des probabilités avec les utilités via aussi un  $\times$ . Il montre enfin une *synthèse* du résultat de l'agrégation via un  $+$  en ce qui concerne les variables d'*environnement* et un  $\max$  en ce qui concerne les variables de *décision*.

## 2.3. Une troisième version impliquant des observations

Supposons que Jean ait la possibilité d'écouter à l'une des trois portes et ainsi d'entendre le gangster, dont il sait qu'il est enrhumé et tousse beaucoup. Supposons de plus que la probabilité de l'entendre soit de 0.8 si on écoute à la porte où il se trouve, de 0.4 si on écoute à une porte adjacente et de 0 sinon.

Pour modéliser ce problème, deux variables supplémentaires sont nécessaires : une variable de décision  $ej$  de domaine  $\{A, B, C\}$  qui représente la porte à laquelle Jean décide d'écouter et une variable d'environnement  $oj$  de domaine  $\{O, N\}$  qui représente le fait que Jean entende le gangster ou non.

Il est aussi nécessaire d'introduire une fonction locale de plausibilité supplémentaire  $P_4$  reliant  $e_j$ ,  $ga$  et  $oj$ , qui exprime la distribution de probabilité conditionnelle sur  $oj$  en fonction de  $e_j$  et de  $ga$ , classiquement notée  $P(oj|e_j, ga)$ . Cette distribution est supposée normalisée ( $\forall e_j, ga, \sum_{oj} P_4 = 1$ ) pour traduire le fait que la disjonction de toutes les situations possibles est certaine.

Dans cette situation avec incertitudes, mais possibilité d'observation, une requête classique est : *Quelle est ou quelles sont la(les) décision(s) de Jean qui maximise(nt) son utilité espérée, si l'on suppose qu'il choisit une porte où écouter et ensuite une porte à ouvrir en fonction de ce qu'il a entendu ?*

Pour répondre à une telle requête, une approche classique consiste à construire un arbre de décision. Dans cet arbre, les variables sont considérées dans un ordre cohérent avec l'ordre des décisions et des observations, par exemple  $\{e_j, oj, ou, ga, tr\}$ . Tout nœud  $n$  de cet arbre correspond à une variable  $x$  et toute arête vers un nœud fils à une affectation  $x = a$ . Si  $x$  est une variable d'environnement, cette arête est de plus pondérée par la probabilité  $P(x = a|A)$ , où  $A$  est l'affectation associée au chemin de la racine à  $n$ . Toute feuille de cet arbre correspond à une affectation  $A$  de l'ensemble des variables et son utilité est l'utilité associée à  $A$ . L'utilité d'un nœud de décision est l'utilité maximale de ses nœuds fils, avec possibilité de mémoriser la(les) décision(s) correspondante(s). L'utilité d'un nœud d'environnement est la somme des utilités de ses nœuds fils, pondérées par le poids des arêtes associées. L'utilité associée à la requête est celle du nœud racine. Il peut cependant être montré (voir la section 3.3.5) que cette approche à base d'arbre de décision est équivalente au calcul de la formule suivante qui ne fait intervenir que les probabilités présentes dans la définition du problème et pas des probabilités de type  $P(x = a|A)$  dont le calcul est potentiellement complexe :

$$\max_{e_j} \sum_{oj} \max_{ou} \sum_{tr, ga} \left( \prod_{i=1}^4 P_i \right) \times \left( \sum_{i=1}^2 U_i \right) \quad [3]$$

À noter que cette équivalence n'est pas limitée au calcul de l'utilité espérée maximale. Elle inclut la mémorisation possible des décisions associées.

Cet exemple montre qu'on aboutit à une *séquence de synthèses* via des + ou des max, avec toujours un + pour les variables d'*environnement* et un max pour les variables de *décision*.

#### 2.4. Une quatrième version impliquant plusieurs agents

*Supposons maintenant que Jean travaille en équipe avec Pierre et que chacun puisse choisir une porte où écouter.*

Cela nous amène à introduire deux variables supplémentaires : une variable de décision  $ep$  de domaine  $\{A, B, C\}$  qui représente la porte à laquelle Pierre décide d'écouter et une variable d'environnement  $op$  de domaine  $\{O, N\}$  qui représente le fait que Pierre entende le gangster ou non. Il est aussi nécessaire d'introduire une fonction locale de plausibilité supplémentaire  $P_5$  reliant  $ep$ ,  $ga$  et  $op$ , similaire à  $P_4$ , qui exprime la distribution de probabilité conditionnelle sur  $op$  en fonction de  $ep$  et de  $ga$ .

Une requête possible est : *Quelle est ou quelles sont la(les) décision(s) de Jean et de Pierre qui maximise(nt) leur utilité espérée, si l'on suppose qu'ils poursuivent tous deux le même objectif, que Pierre choisisse une porte où écouter, qu'il transmette à Jean l'information recueillie et que Jean choisisse alors une porte où écouter, puis une porte à ouvrir en fonction de l'information recueillie ?*

De même que précédemment, il est possible de montrer (voir la section 3.3.5) qu'une approche à base d'arbre de décision est équivalente au calcul de la formule suivante :

$$\max_{ep} \sum_{op} \max_{ej} \sum_{oj} \max_{ou} \sum_{tr,ga} \left( \left( \prod_{i=1}^5 P_i \right) \times \left( \sum_{i=1}^2 U_i \right) \right) \quad [4]$$

*Mais supposons maintenant que Pierre soit un faux ami et que son objectif soit de minimiser l'utilité de Jean.*

On peut montrer que la formule associée est simplement la précédente où on a remplacé  $\max_{ep}$  par  $\min_{ep}$  :

$$\min_{ep} \sum_{op} \max_{ej} \sum_{oj} \max_{ou} \sum_{tr,ga} \left( \left( \prod_{i=1}^5 P_i \right) \times \left( \sum_{i=1}^2 U_i \right) \right) \quad [5]$$

*Supposons en plus que Pierre ne communique pas à Jean l'information obtenue en écoutant.*

On peut montrer que la formule associée est la précédente où on a déplacé  $\sum_{op}$  dans la séquence de synthèses pour le placer après  $\max_{ou}$  :

$$\min_{ep} \max_{ej} \sum_{oj} \max_{ou} \sum_{op, tr, ga} \left( \left( \prod_{i=1}^5 P_i \right) \times \left( \sum_{i=1}^2 U_i \right) \right) \quad [6]$$

Ces différents exemples montrent qu'on aboutit toujours à une *séquence de synthèses* via des +, des max ou des min, avec un + pour les variables d'environnement et un max ou un min pour les variables de décision, suivant le caractère coopératif ou non de l'agent décideur.

## 2.5. Une cinquième version impliquant des faisabilités

*Supposons maintenant que Jean et Pierre ne puissent pas écouter à la même porte et qu'en plus la porte C soit bloquée.*

Ces contraintes sur les décisions peuvent être modélisées par deux fonctions locales de faisabilité : une première  $F_1$  qui exprime que Jean et Pierre ne peuvent pas écouter à la même porte (contrainte dure  $ej \neq ep$ ) et une seconde  $F_2$  qui exprime que la porte  $C$  est bloquée (contrainte dure  $ou \neq C$ ). De la même façon qu'avec les plausibilités, ces contraintes sont supposées exprimer des distributions de faisabilité normalisées ( $\forall_{ej,ep} F_1 = t$  et  $\forall_{ou} F_2 = t$ ) pour traduire le fait qu'une décision est toujours possible, même s'il s'agit de ne rien faire.

La prise en compte de ces contraintes sur les décisions nous conduit à introduire un élément spécial, noté  $\diamond$ , représentant l'infaisabilité, avec comme particularités le fait qu'il soit absorbant pour tout opérateur d'agrégation (l'agrégation de quelque chose d'infaisable avec n'importe quoi est infaisable) et neutre pour tout opérateur de synthèse (ce qui est infaisable est de toute façon ignoré). Elle nous conduit aussi à introduire un opérateur spécial, noté  $\star$ , permettant d'agréger les faisabilités avec les utilités espérées et vérifiant la propriété suivante :  $\forall u, (f \star u = \diamond) \wedge (t \star u = u)$ . Avec ces notations, on peut montrer que la formule 6 devient :

$$\min_{ep} \max_{ej} \sum_{oj} \max_{ou} \sum_{op, tr, ga} ((\bigwedge_{i=1}^2 F_i) \star ((\prod_{i=1}^5 P_i) \times (\sum_{i=1}^2 U_i))) \quad [7]$$

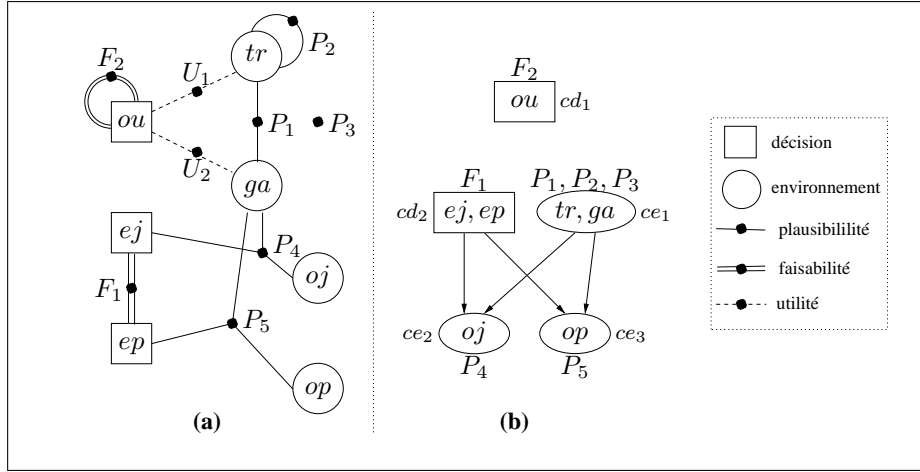
Cet exemple montre l'agrégation des utilités via un  $+$ , des probabilités via un  $\times$ , des faisabilités via un  $\wedge$  logique, des probabilités et des utilités via un  $\times$  et enfin des faisabilités et des utilités espérées via l'opérateur spécial  $\star$ .

Il est possible de bâtir une représentation graphique du problème ainsi modélisé (voir la figure 1(a)) sous la forme d'un *hyper-graphe* où les nœuds représentent les *variables* (variables de décisions dans un carré et variables d'environnement dans un cercle) et où les hyper-arêtes représentent les portées des *fonctions locales* (fonctions locales d'utilité avec des lignes pointillées, fonctions locales de plausibilité avec des lignes pleines et fonctions locales de faisabilité avec des doubles lignes).

Il est cependant possible de bâtir une autre représentation, complémentaire de la précédente (voir la figure 1(b)) sous la forme d'un *graphe orienté acyclique* où les nœuds représentent des *composantes* (ensembles de variables de même nature, soit décision, soit environnement, corrélées par des *relations de dépendance non orientées* ; composantes de décision dans un rectangle et composantes d'environnement dans une ellipse) et où les arcs représentent des *relations de dépendance orientées*. Les variables de décision  $ej$  et  $ep$  sont par exemple corrélées par la fonction locale de faisabilité  $F_1$ . De la même façon, les variables d'environnement  $ga$  et  $tr$  sont corrélées par les fonctions locales de plausibilité  $P_1$ ,  $P_2$  et  $P_3$ . Par ailleurs, un arc apparaît de



la composante  $cd_2$  vers la composante  $ce_2$  du fait de la dépendance de  $oj$  par rapport à  $ej$ . Une autre arc apparaît de la composante  $ce_1$  vers la composante  $ce_2$  du fait de la dépendance de  $oj$  par rapport à  $ga$ . Cette seconde représentation fait apparaître les *dépendances* entre composantes (voir la section 3.2.5 pour une interprétation plus formelle de cette représentation).



**Figure 1:** Représentation graphique de l'exemple (a) hyper-graphe des fonctions locales entre variables (b) graphe orienté acyclique entre composantes.

## 2.6. D'autres opérateurs pour les plausibilités et les utilités

L'exemple précédent utilise des *probabilités* pour modéliser l'incertitude. Sous hypothèse d'indépendance, les probabilités sont agrégées via un opérateur  $\otimes_p = \times$  et synthétisées via un opérateur  $\oplus_p = +$ . Mais d'autres théories peuvent être utilisées pour modéliser l'incertitude, comme par exemple la théorie des *possibilités* (Dubois *et al.*, 1988) ou celle des *fonctions de Spohn* (Spohn, 1990). Avec la première, une option possible consiste à utiliser  $\otimes_p = \min$  et  $\oplus_p = \max$ , tandis qu'avec la seconde,  $\otimes_p = +$  et  $\oplus_p = \min$ .

De façon similaire, l'exemple précédent utilise des *utilités additives* (des coûts ou des gains) pour modéliser les préférences. Elles sont agrégées via un opérateur  $\otimes_u = +$ . Mais si les préférences sont modélisées par des *priorités*, elles sont agrégées via  $\otimes_u = \min$ .

Enfin, dans la théorie de l'*utilité espérée probabiliste*, la formule classique  $\sum_i p_i \times u_i$  traduit une agrégation des probabilités et des utilités via un opérateur  $\otimes_{pu} = \times$  et une synthèse du résultat via un opérateur  $\oplus_u = +$ . Mais cette formule se généralise sous la forme  $\oplus_u (p_i \otimes_{pu} u_i)$ , avec par exemple  $\otimes_{pu} = \min$  et  $\oplus_u = \max$  dans la théorie de l'*utilité espérée possibiliste optimiste* ou  $\otimes_{pu} = +$

et  $\oplus_u = \min$  dans la théorie de l'utilité associée aux *fonctions de Spohn* (avec des utilités uniquement positives).

Le tableau 1 résume les ensembles utilisés pour exprimer plausibilités et utilités ( $E_p$  et  $E_u$ ), ainsi que les opérateurs utilisés pour les agréger ( $\otimes_p$ ,  $\otimes_u$  et  $\otimes_{pu}$ ) et les synthétiser ( $\oplus_p$  et  $\oplus_u$ ), dans sept cadres théoriques classiques, sachant que d'autres cadres existent au delà de ces sept-là.

	$E_p$	$\oplus_p$	$\otimes_p$	$E_u$	$\otimes_u$	$\oplus_u$	$\otimes_{pu}$
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	+
6	$\{t, f\}$	$\vee$	$\wedge$	$\{t, f\}$	$\wedge$	$\vee$	$\wedge$
7	$\{t, f\}$	$\vee$	$\wedge$	$\{t, f\}$	$\wedge$	$\wedge$	$\rightarrow$

**Tableau 1.** Ensembles et opérateurs utilisés dans sept cadres classiques : (1) utilité espérée quantitative probabiliste, (2) satisfaction espérée quantitative probabiliste, (3) utilité espérée qualitative possibiliste optimiste, (4) utilité espérée qualitative possibiliste pessimiste, (5) utilité qualitative avec fonctions de Spohn et utilités uniquement positives, (6) satisfaction espérée booléenne optimiste, (7) satisfaction espérée booléenne pessimiste.

En utilisant ces notations, la formule 7 prend la forme suivante, avec  $F$  l'ensemble des relations locales de faisabilité,  $P$  l'ensemble des fonctions locales de plausibilité et  $U$  l'ensemble des fonctions locales d'utilité :

$$\min_{ep} \max_{ej} \oplus_u \max_{oj} \oplus_u \max_{op, tr, ga} ((\bigwedge_{F_i \in F} F_i) \star ((\otimes_p P_i) \otimes_{pu} (\otimes_u U_i))) \quad [8]$$

Cette formule montre la forme générale d'une requête dans le cadre PFU que nous allons maintenant formellement introduire.

### 3. Le cadre PFU

Le cadre PFU a trois composants essentiels : (1) une *structure algébrique générique* qui capture la façon d'exprimer, d'agréger et de synthétiser des plausibilités, des faisabilités et des utilités, (2) une *structure graphique générique* qui permet de décrire des variables, des composantes et des fonctions locales de plausibilité, de faisabilité et d'utilité et (3) une *structure générique des requêtes* qui capture la façon de synthétiser l'information au niveau des différentes variables. Nous les présentons successivement.

### 3.1. Une structure algébrique générique

La structure algébrique que nous utilisons s'appuie sur les travaux de Friedman, Halpern et Chu (Friedman *et al.*, 1995; Halpern, 2001; Chu *et al.*, 2003), qui proposent une généralisation axiomatique et algébrique des notions classiques de probabilité, utilité et utilité espérée, sous les dénominations de *plausibilité*, d'*utilité* et d'*utilité espérée généralisée*. La structure proposée s'écarte cependant sur certains points de leurs propositions.

#### 3.1.1. Une structure algébrique de plausibilité

Une *structure de plausibilité* est définie 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*<sup>1</sup> et  $\otimes_p$  un *opérateur d'agrégation* sur les plausibilités. Nous imposons à cette structure de satisfaire les axiomes de base sur les plausibilités proposés dans (Halpern, 2001). Nous étendons cependant la structure proposée par Halpern pour que  $\oplus_p$  et  $\otimes_p$  soient clos sur  $E_p$ . Ceci nous conduit à la définition suivante :

**Définition 1** Une *structure de plausibilité* est un triplet  $(E_p, \oplus_p, \otimes_p)$  tel que :

- $(E_p, \oplus_p, \otimes_p)$  est un *semi-anneau commutatif*, ce qui implique que les opérateurs binaires  $\oplus_p$  et  $\otimes_p$  ( $E_p \times E_p \rightarrow E_p$ ) soient *associatifs et commutatifs* et possèdent chacun un *élément neutre*, noté  $0_p$  pour  $\oplus_p$  et  $1_p$  pour  $\otimes_p$ , que  $0_p$  soit *absorbant* pour  $\otimes_p$  et que  $\otimes_p$  soit *distributif* par rapport à  $\oplus_p$  ;
- $E_p$  est équipé d'un *ordre partiel*  $\preceq_p$  dont  $0_p$  est l'*élément minimum* ;
- $\oplus_p$  et  $\otimes_p$  sont *monotones* pour  $\preceq_p$ .

$0_p$  sera associé aux événements *impossibles* et  $1_p$  aux événements *certaines*. À noter qu'alors que cette définition impose que  $0_p$  soit l'élément minimum de  $E_p$ , elle n'impose pas que  $1_p$  en soit l'élément maximum. La structure de plausibilité associée aux probabilités est par exemple  $(\mathbb{R}^+, +, \times)$ , avec  $\preceq_p = \leq$ ,  $0_p = 0$  et  $1_p = 1$ . Toutes les structures  $(E_p, \oplus_p, \otimes_p)$  présentées dans le tableau 1 sont des structures de plausibilité.

#### 3.1.2. Une structure algébrique de faisabilité

Du fait qu'une décision est soit faisable, soit infaisable, la *structure de faisabilité* que nous utilisons n'est pas paramétrable. C'est en fait un cas particulier de structure de plausibilité :  $(\{t, f\}, \vee, \wedge)$  avec  $f \preceq_p t$ ,  $0_p = f$  et  $1_p = t$ .

---

1. Nous avons jusqu'à présent utilisé le terme informel de *synthèse*. À partir de maintenant, nous employons le terme équivalent mais plus formel d'*élimination* : soit une fonction locale  $L_i$  de portée  $S(L_i)$ , une variable  $x \in S(L_i)$  de domaine  $Dom(x)$  et un opérateur d'élimination  $op$  ; le résultat de l'élimination par  $op$  de  $x$  de  $L_i$ , noté  $op_x L_i$ , est une fonction locale de portée  $S(L_i) - \{x\}$  ( $x$  a été éliminé), telle que  $\forall A \in S(L_i) - \{x\}$ ,  $(op_x L_i)(A) = op_{a \in Dom(x)} L_i(A.(x, a))$ .

### 3.1.3. Une structure algébrique d'utilité

Une *structure d'utilité* est définie 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  $\otimes_u$  un *opérateur d'agrégation* sur les utilités. Des axiomes classiques sur les utilités nous conduisent à la définition suivante :

**Définition 2** Une *structure d'utilité* est une paire  $(E_u, \otimes_u)$  telle que :

- $(E_u, \otimes_u)$  est un monoïde commutatif, ce qui implique que l'opérateur binaire  $\otimes_u (E_u \times E_u \rightarrow E_u)$  soit associatif et commutatif et possède un élément neutre noté  $1_u$  ;
- $E_u$  est équipé d'un ordre partiel  $\preceq_u$ .

$1_u$  sera associé aux situations *indifférentes* du point de vue de l'agrégation des utilités. La structure d'utilité associée aux utilités additives classiques est par exemple  $(\mathbb{R}, +)$ , avec  $\preceq_u = \leq$  et  $1_u = 0$ . Toutes les structures  $(E_u, \otimes_u)$  présentées dans le tableau 1 sont des structures d'utilité.

### 3.1.4. Une structure algébrique d'utilité espérée

Une *structure d'utilité espérée* est définie comme un quadruplet  $(E_p, E_u, \oplus_u, \otimes_{pu})$  où  $\oplus_u$  est un opérateur d'élimination sur les utilités et  $\otimes_{pu}$  un opérateur d'agrégation entre plausibilités et utilités. Nous imposons à cette structure de satisfaire les axiomes de base sur l'utilité espérée généralisée proposés dans (Chu *et al.*, 2003). Pour prendre en compte l'aspect *séquentiel* de la décision, nous imposons cependant des axiomes supplémentaires justifiés par la *théorie des loteries* (von Neumann *et al.*, 1944). Ceci nous conduit à la définition suivante :

**Définition 3** Une *structure d'utilité espérée* est un quadruplet  $(E_p, E_u, \oplus_u, \otimes_{pu})$  tel que :

-  $(E_u, \oplus_u, \otimes_{pu})$  est un *semi-module* sur  $(E_p, \oplus_p, \otimes_p)$ , ce qui implique que  $(E_p, \oplus_p, \otimes_p)$  soit un *semi-anneau commutatif* (voir la définition 1), que  $(E_u, \oplus_u)$  soit un *monoïde commutatif* (voir la définition 2) avec un élément neutre noté  $0_u$  et que l'opérateur binaire  $\otimes_{pu} (E_p \times E_u \rightarrow E_u)$  satisfasse les propriétés suivantes :

- *distributivité par rapport à  $\oplus_p$  et  $\oplus_u$  :*
- $\forall p_1, p_2 \in E_p, \forall u \in E_u, p_1 \otimes_{pu} (p_2 \otimes_{pu} u) = (p_1 \otimes_p p_2) \otimes_{pu} u$  ;
- $\forall u \in E_u, (0_p \otimes_{pu} u = 0_u) \wedge (1_p \otimes_{pu} u = u)$  ;
- $\oplus_u$  est *monotone* et  $\otimes_{pu}$  *monotone à droite* pour  $\preceq_u$ .

$0_u$  sera associé aux situations *indifférentes* du point de vue de l'élimination des utilités. La structure d'utilité espérée associée à l'utilité espérée probabiliste est par exemple  $(\mathbb{R}^+, \mathbb{R}, +, \times)$ , avec  $0_u = 0$ . Toutes les structures présentées dans le tableau 1 sont des structures d'utilité espérée.

### 3.1.5. Hypothèses implicites et cadres non couverts

Les hypothèses faites sur les structures de plausibilité, d'utilité et d'utilité espérée résultent de compromis toujours discutables entre des considérations sémantiques et algorithmiques : pouvoir englober le plus grand nombre possible de situations et d'approches, garder à l'esprit que les problèmes concrets doivent pouvoir être représentés de façon suffisamment compacte et résolu de façon raisonnablement efficace.

Par exemple, le fait de supposer l'existence d'ensembles  $E_p$  et  $E_u$  de degrés de plausibilité et d'utilité impose que plausibilités et utilités soient *cardinales*. Des approches purement ordinales comme par exemple les réseaux de préférences conditionnelles (*CP-nets* (Boutilier *et al.*, 1999)) ne sont pas couvertes par le cadre proposé.

De la même façon, le fait que l'agrégation d'une plausibilité avec une utilité produise une utilité ( $\otimes_{pu} : E_p \times E_u \rightarrow E_u$ ) repose sur une hypothèse implicite de *commensurabilité* entre plausibilités et utilités. Des travaux tels que (Fargier *et al.*, 1999), qui ne font pas cette hypothèse, ne sont pas couverts.

Enfin, les axiomes imposés impliquent que seules des plausibilités à caractère *distributionnel* sont couvertes : la plausibilité d'un ensemble  $E$  d'affectations des variables est supposée être complètement déterminée par la plausibilité de chacune des affectations complètes couvertes par  $E$ . De ce fait, des cadres non distributionnels comme les *fonctions de croyance* de Dempster-Shafer (Shafer, 1976) ne sont pas couverts.

## 3.2. Une structure graphique générique

### 3.2.1. Variables

Les exemples de la section 2 mettent en évidence une distinction importante entre variables de décision, contrôlées par un agent décideur, et variables d'environnement, non contrôlées par un agent décideur.

En conséquence, le premier élément de la structure graphique proposée est un ensemble  $V$  de *variables à domaines finis*, partitionné en deux sous-ensembles : un ensemble  $V_D$  de *variables de décision* ( $ej$ ,  $ep$  et  $ou$  dans l'exemple) et un ensemble  $V_E$  de *variables d'environnement* ( $tr$ ,  $ga$ ,  $oj$  et  $op$  dans l'exemple).

### 3.2.2. Composantes

Les mêmes exemples mettent en évidence des relations de natures différentes entre variables. Certaines de ces relations traduisent des *corrélations* entre variables. Par exemple, les variables de décision  $ej$  et  $ep$  sont corrélées via la relation de faisabilité  $F_1$  (Jean et Pierre ne peuvent pas écouter à la même porte). De la même façon, les variables  $tr$  et  $ga$  sont corrélées via la fonction de plausibilité  $P_1$  (le trésor et le gangster ne se trouvent pas derrière la même porte). D'autres relations traduisent des *dépendances causales* entre variables. Par exemple, la valeur de  $oj$  dépend de celles

de  $ej$  et  $ga$  via la fonction de plausibilité  $P_4$  (ce que Jean entend dépend de la porte à laquelle il écoute et de la porte derrière laquelle se trouve le gangster).

Pour prendre ceci en compte, le second élément de la structure graphique proposée est une partition de  $V_D$  en *composantes de décision* et une partition de  $V_E$  en *composantes d'environnement*. On obtient un ensemble  $C$  de composantes, lui-même partitionné en deux sous-ensembles : un ensemble  $C_D$  de composantes de décision ( $cd_1 = \{ou\}$  et  $cd_2 = \{ej, ep\}$  dans l'exemple) et un ensemble  $C_E$  de composantes d'environnement ( $ce_1 = \{tr, ga\}$ ,  $ce_2 = \{oj\}$  et  $ce_3 = \{op\}$  dans l'exemple). On introduit ensuite un *graphe orienté acyclique* (DAG) sur  $C$  (voir la figure 1b). Implicitement, les variables d'une même composante sont corrélées par des relations de dépendance *non orientées* et le DAG exprime des relations de dépendance *orientées* entre composantes. Ces dernières peuvent (sans que cela soit obligatoire) traduire des relations de *causalité*, comme celles existant de  $ej$  et  $ga$  vers  $oj$ .

### 3.2.3. Fonctions locales

Les exemples de la section 2 montrent aussi trois types de *fonctions locales* : fonctions locales de *plausibilité*, de *faisabilité* et d'*utilité*.

Implicitement, les fonctions de plausibilité sur des variables d'environnement d'une même composante expriment des corrélations entre ces variables (par exemple,  $P_1$  sur  $tr$  et  $ga$ ). De même, les fonctions de faisabilité sur des variables de décision d'une même composante expriment des corrélations entre ces variables (par exemple,  $F_1$  sur  $ej$  et  $ep$ ). Les fonctions de plausibilité sur des variables d'environnement et de décision de composantes différentes expriment le fait que des variables d'environnement dépendent d'autres variables d'environnement et de décision (par exemple,  $P_4$  qui exprime que  $oj$  dépend de  $ej$  et  $ga$ ). De même, les fonctions de faisabilité sur des variables d'environnement et de décision de composantes différentes expriment le fait que des variables de décision dépendent d'autres variables de décision et d'environnement. Finalement, les fonctions locales d'utilité portent indifféremment sur les variables de décision ou d'environnement.

### 3.2.4. Définition d'un PFU

Ceci nous conduit à la définition suivante d'un PFU :

**Définition 4** *Un PFU est un triplet  $(V, G, L)$  tel que :*

- $V$  est un ensemble de variables à domaines finis, partitionné en un ensemble  $V_D$  de variables de décision et un ensemble  $V_E$  de variables d'environnement ; pour tout  $V' \subseteq V$ , on note  $Dom(V')$  le produit cartésien des domaines de valeur des variables de  $V'$  ;

- $G$  est un *graphe orienté acyclique* (DAG) dont les sommets, appelés *composantes*, sont une partition  $C$  de  $V$ , plus précisément l'union d'une partition  $C_D$  de  $V_D$  et d'une partition  $C_E$  de  $V_E$  ; pour tout  $c \in C$ , on note  $pa_G(c)$  l'union des *composantes parentes* de  $c$  dans  $G$  ;

–  $L$  est un ensemble de fonctions locales, partitionné en un ensemble  $P$  de fonctions locales de plausibilité, un ensemble  $F$  de fonctions locales de faisabilité et un ensemble  $U$  de fonctions locales d'utilité ; pour toute fonction locale  $L_i$ , on note  $S(L_i)$  l'ensemble des variables sur lesquelles elle est définie :

-  $P = \{P_1, P_2, \dots\}$  est un ensemble de fonctions locales de plausibilité, une fonction locale de plausibilité  $P_i$  étant simplement une fonction de  $\text{Dom}(S(P_i))$  dans  $E_p$  ; toute fonction locale de plausibilité  $P_i$  est associée à une composante d'environnement unique  $c \in C_E$ , notée  $c(P_i)$ , telle que  $S(P_i) \subset c \cup \text{pa}_G(c)$  ; de plus, pour toute composante d'environnement  $c \in C_E$ , la condition de normalisation suivante est satisfaite pour tout élément de  $\text{Dom}(\text{pa}_G(c))$  :  $\bigoplus_{\text{Dom}(c)} \left( \bigotimes_{c(P_i)=c} P_i \right) = 1_p$ , exprimant le fait que la disjonction de toutes les situations possibles est certaine ;

-  $F = \{F_1, F_2, \dots\}$  est un ensemble de fonctions locales de faisabilité, une fonction locale de faisabilité  $F_i$  étant simplement une fonction de  $\text{Dom}(S(F_i))$  dans  $\{t, f\}$  ; toute fonction locale de faisabilité  $F_i$  est associée à une composante de décision unique  $c \in C_D$ , notée  $c(F_i)$ , telle que  $S(F_i) \subset c \cup \text{pa}_G(c)$  ; de plus, pour toute composante de décision  $c \in C_D$ , la condition de normalisation suivante est satisfaite pour tout élément de  $\text{Dom}(\text{pa}_G(c))$  :  $\bigvee_{\text{Dom}(c)} \left( \bigwedge_{c(F_i)=c} F_i \right) = t$ , exprimant le fait que la disjonction de toutes les décisions possibles est faisable ;

-  $U = \{U_1, U_2, \dots\}$  est un ensemble de fonctions locales d'utilité, une fonction locale d'utilité  $U_i$  étant simplement une fonction de  $\text{Dom}(S(U_i))$  dans  $E_u$ .

Sur l'exemple de la figure 1b, les conditions de normalisation des plausibilités et des faisabilités imposent que :

$$\bigvee_{ou} F_2 = t, \bigvee_{ej,ep} F_1 = t, \sum_{tr,ga} \left( \prod_{i=1}^3 P_i \right) = 1, \forall_{ej,ep,tr,ga}, \sum_{oj} P_4 = \sum_{op} P_5 = 1$$

À un PFU, on peut donc associer (1) un *hyper-graphe* dont les nœuds représentent les variables et les hyper-arêtes les portées des fonctions locales (voir la figure 1a) et (2) un *graphe orienté acyclique* dont les nœuds représentent les composantes et les arcs les dépendances entre composantes (voir la figure 1b).

### 3.2.5. Indépendance conditionnelle et factorisation

Sur cette base, il est possible d'étendre un résultat central sur les réseaux bayésiens (Pearl, 1988), qui fait le lien entre le DAG associé à un réseau bayésien et la factorisation de la distribution de probabilité jointe exprimée par le réseau. Ce résultat utilise la notion de *compatibilité* entre un DAG et une distribution de probabilité jointe : un DAG  $G$  sur un ensemble de variables  $V$  est dit compatible avec une distribution de probabilité jointe sur  $V$  si et seulement si chaque variable  $v \in V$  est conditionnellement indépendante de ses non-descendants dans  $G$ , étant donnés ses parents dans  $G$ . Ceci conduit au théorème suivant : si un DAG  $G$  sur un ensemble de

variables  $V$  est compatible avec une distribution de probabilité jointe  $\mathcal{P}_V$  sur  $V$ , alors  $\mathcal{P}_V = \prod_{v \in V} \mathcal{P}_{v|pa_G(v)}$ .

L'extension de ce résultat probabiliste au cadre plus général des plausibilités nécessite d'étendre la notion d'*indépendance conditionnelle*. Encore une fois, nous nous appuyons sur les travaux de Friedman et Halpern (Friedman *et al.*, 1995; Halpern, 2001), tout en nous en écartant sur certains aspects. Si  $V$  est un ensemble de variables, nous définissons une *distribution de plausibilité* sur  $V$  comme une fonction  $\mathcal{P}_V$  de  $Dom(V)$  dans  $E_p$ , telle que  $\bigoplus_{A \in Dom(V)} \mathcal{P}_V(A) = 1_p$ . Une distribution de plausibilité sur  $V$  induit une distribution de plausibilité sur tout  $V' \subseteq V$  :  $\mathcal{P}_{V'} = \bigoplus_{Dom(V-V')} \mathcal{P}_V$ .

L'introduction d'un opérateur de conditionnement  $\otimes_p$  vérifiant un certain nombre de propriétés (pour plus de précisions, voir (Pralet *et al.*, 2005)) permet de définir une *distribution de plausibilité conditionnelle* sur  $V_1$  sachant  $V_2$  ( $V_1$  et  $V_2$  disjoints) :  $\mathcal{P}_{V_1|V_2} = \mathcal{P}_{V_1, V_2} \otimes_p \mathcal{P}_{V_2}$ . Dans le cas des probabilités, par exemple,  $\otimes_p = /$ . Un *opérateur de conditionnement* respectant les propriétés requises peut être défini sur toutes les structures présentées dans le tableau 1. Sur cette base, nous disons que  $V_1$  est *conditionnellement indépendant* de  $V_2$  sachant  $V_3$  ( $V_1$ ,  $V_2$  et  $V_3$  disjoints) si et seulement si  $\mathcal{P}_{V_1, V_2|V_3} = \mathcal{P}_{V_1|V_3} \otimes_p \mathcal{P}_{V_2|V_3}$ . Nous disons aussi qu'un DAG  $G$  sur un ensemble de composantes  $C$  est compatible avec une distribution de plausibilité jointe sur  $V = \cup_{c \in C} c$  si et seulement si chaque composante  $c \in G$  est conditionnellement indépendante de ses non-descendants dans  $G$ , étant données ses parentes dans  $G$ . Il est alors possible d'établir le théorème suivant :

**Théorème 1** *Si un DAG  $G$  sur un ensemble de composantes  $C$  est compatible avec une distribution de plausibilité  $\mathcal{P}_V$  jointe sur  $V = \cup_{c \in C} c$ , alors  $\mathcal{P}_V = \bigotimes_{c \in C} \mathcal{P}_{c|pa_G(c)}$ .*

Le théorème 1 nous fournit un premier niveau de factorisation composante par composante de la distribution de plausibilité jointe. Mais il est possible d'aller plus loin puisque, pour chaque composante  $c \in C$ , la distribution de plausibilité conditionnelle  $\mathcal{P}_{c|pa_G(c)}$  peut être elle-même factorisée en fonctions locales de plausibilité  $P_i$  associées à  $c$  ( $\mathcal{P}_{c|pa_G(c)} = \bigotimes_{c(P_i)=c} P_i$ ) à condition que, pour tout élément de  $Dom(pa_G(c))$ ,  $\bigoplus_{Dom(c)} (\bigotimes_{c(P_i)=c} P_i) = 1_p$  : le résultat de l'agrégation des fonctions locales est bien une distribution de plausibilité conditionnelle. Par exemple, dans le cas d'un réseau bayésien, rien ne s'oppose à ce que les probabilités conditionnelles  $\mathcal{P}_{v|pa_G(v)}$  s'expriment comme un produit de fonctions plus locales, éventuellement sous forme de contraintes en cas d'information binaire (0 ou 1 ; voir par exemple (Dechter *et al.*, 2001)).

Ceci montre que, de même qu'un réseau bayésien est une façon compacte d'exprimer une distribution de probabilité jointe sur un ensemble de variables, un PFU est une façon compacte d'exprimer une distribution de plausibilité jointe sur les variables d'environnement ( $V_E$ ) (plus précisément une distribution de plausibilité jointe



sur les variables d'environnement, conditionnée par les valeurs des variables de décision ( $V_D$ )).

Comme la structure de faisabilité est un cas particulier de structure de plausibilité, le même type de résultat peut être établi en ce qui concerne les faisabilités : un PFU est une façon compacte d'exprimer une distribution de faisabilité jointe sur  $V_D$  (plus précisément sur  $V_D$  sachant  $V_E$ ).

Ceci nous montre finalement une façon naturelle de construire le PFU correspondant à un problème concret : identifier les variables de décision et d'environnement, identifier les composantes et construire le DAG sur la base des indépendances conditionnelles, exprimer les distributions de plausibilité ou de faisabilité conditionnelles associées à chaque composante via des fonctions aussi locales que possible, exprimer enfin les fonctions locales d'utilité.

### 3.2.6. Cadres couverts

Le cadre des PFU couvre de nombreux cadres existant en Intelligence Artificielle, en particulier tous les cadres énumérés au début de la section 1.

Par exemple, un *problème de satisfaction de contraintes* (CSP)  $P$  peut être modélisé comme un PFU  $(V, G, L)$  où  $V$  contient toutes les variables de  $P$ , considérées comme des variables de décision, où  $G$  contient une seule composante englobant toutes les variables et où  $L$  contient toutes les contraintes de  $P$ , considérées comme des fonctions locales d'utilité (si elles étaient considérées comme des fonctions locales de faisabilité, les conditions de normalisation interdiraient de considérer des CSP incohérents). La démarche est identique pour un *problème de satisfiabilité d'une formule booléenne* (SAT), à la seule différence que les variables sont booléennes et les contraintes des clauses. Il en est de même pour les CSP *valués* et, malgré les apparences, pour les *formules booléennes quantifiées* (QBF) et les CSP *quantifiés* (QCSP) : les différences apparaîtront au niveau des requêtes ; voir la section 3.3. Avec les CSP *mixtes*, la distinction est faite entre variables de décision et variables d'environnement. Avec les *problèmes de satisfiabilité stochastiques* (SSAT) et les CSP *stochastiques* (SCSP), des composantes apparaissent et des fonctions locales de plausibilité s'ajoutent aux fonctions locales d'utilité.

Un *réseau bayésien*  $P$  peut être modélisé comme un PFU  $(V, G, L)$  où  $V$  contient toutes les variables de  $P$ , considérées comme des variables d'environnement, où une composante est associée à chaque variable, où le DAG  $G$  est celui de  $P$  et où  $L$  contient les distributions de probabilité conditionnelle de  $P$  (une par variable), considérées comme des fonctions locales de plausibilité.

Un *diagramme d'influence*  $P$  peut aussi être modélisé comme un PFU  $(V, G, L)$  :  $V$  contient les variables d'environnement et les variables de décision de  $P$  ; une composante est associée à chaque variable ;  $G$  est le DAG associé à  $P$  sans les variables d'utilité et sans les arcs vers les variables de décision et d'utilité ;  $L$  contient, d'une part, des fonctions locales de plausibilité correspondant aux distributions de probabilité conditionnelle sur les variables d'environnement et, d'autre part, des fonctions

locales d'utilité (une associée à chaque variable d'utilité  $u$  de  $P$ , dont la portée est l'ensemble des parents de  $u$  dans le DAG associé à  $P$ ). Avec les *réseaux de valuation*, apparaissent en plus des fonctions locales de faisabilité.

Un *processus décisionnel markovien* à horizon fini (MDP)  $P$  peut aussi être modélisé comme un PFU  $(V, G, L)$  : pour chaque instant  $i$ , une variable d'environnement  $s_i$  est associée à l'état à l'instant  $i$  et une variable de décision  $d_i$  à la décision à l'instant  $i$  ; une composante est associée à chaque variable ;  $G$  est le DAG où, pour chaque instant  $i$ ,  $d_i$  a pour parent  $s_i$  et  $s_{i+1}$  a pour parent  $s_i$  et  $d_i$  ;  $L$  contient des fonctions locales de plausibilité correspondant aux probabilités de transition  $P(s_{i+1}|s_i, d_i)$ , des fonctions locales de faisabilité correspondant aux contraintes sur les décisions  $C(d_i|s_i)$  et des fonctions locales d'utilité correspondant aux gains locaux  $G(s_i, d_i, s_{i+1})$ . Pour les *processus partiellement observables* (POMDP), il est nécessaire d'ajouter, pour chaque instant  $i$ , une variable d'environnement  $o_i$  associée à l'observation à l'instant  $i$ , d'indiquer dans le DAG que pour chaque instant  $i$ ,  $o_i$  a pour parent  $s_i$  et d'ajouter les probabilités d'observation  $P(o_i|s_i)$  aux fonctions locales de plausibilité. Avec les MDP *factorisés*, les variables  $s_i$  sont décomposées en autant de variables que de facteurs.

Il en est de même pour un *problème de planification* à horizon fixé  $h$  dans le cadre STRIPS. Il peut être modélisé comme un PFU  $(V, G, L)$  : une variable d'environnement  $s_i$  est associée à chaque instant  $i$  et à chaque composante booléenne de l'état ; une variable de décision  $d_i$  est associée à chaque instant ;  $G$  est le DAG où, pour chaque instant  $i$ ,  $d_i$  a *a priori* pour parents tous les  $s_i$  et  $s_{i+1}$  pour parents  $d_i$  et tous les  $s_i$  ;  $L$  contient des fonctions locales de faisabilité correspondant aux pré-conditions des actions, des fonctions locales de plausibilité correspondant aux effets des actions et une seule fonction d'utilité booléenne correspondant au but et portant sur les variables  $s_h$ .

### 3.3. Une structure générique des requêtes

#### 3.3.1. Définition d'une requête sur un PFU

Les exemples de la section 2 montrent des séquences d'éliminations de variables qui sont à chaque fois cohérentes avec l'ordre des décisions et des observations, des opérateurs d'élimination dépendant de la nature des variables (décision ou environnement) et, en ce qui concerne les variables de décision, dépendant du caractère coopératif ou non des agents. Ceci nous conduit à la définition suivante d'une *requête* sur un PFU :

**Définition 5** Une requête sur un PFU  $(V, G, L)$  est une séquence *Sov* de paires (opérateur d'élimination-ensemble de variables)  $(op, V')$  telles que tous les  $V'$  sont dis-joints et que, soit  $V' \subseteq V_D$  (variables de décision) et  $op = \min$  ou  $\max$ , soit  $V' \subseteq V_E$  (variables d'environnement) et  $op = \oplus_u$ .

Cette définition n'impose pas que toutes les variables de  $V$  apparaissent dans  $Sov$ . Les variables qui n'y apparaissent pas sont dites *libres*, les autres *liées*. On notera que rien n'interdit que, dans certains cadres comme les cinq derniers du tableau 1,  $\oplus_u = \min$  ou  $\oplus_u = \max$ , ce qui veut dire que les mêmes opérateurs  $\min$  ou  $\max$  sont utilisés pour éliminer les variables de décision et les variables d'environnement.

La définition d'une sémantique associée aux requêtes nous impose cependant une condition supplémentaire, liée à la causalité que le DAG peut exprimer. Si l'on prend l'exemple de la formule 3, nous voulons exclure une requête telle que :

$\sum_{oj} \max_{ej} \max_{ou} \sum_{tr,ga} \left( \prod_{i=1}^4 P_i \right) \times \left( \sum_{i=1}^2 U_i \right)$ , où les éliminations de  $ej$  et  $oj$  ont été inversées, car cela signifierait qu'on peut parler de ce que Jean a entendu avant qu'il ait choisi la porte où écouter. Ceci nous conduit à la définition suivante d'une *requête correcte* sur un PFU :

**Définition 6** Une requête  $Sov$  sur un PFU  $(V, G, L)$  est correcte si et seulement, pour toute paire  $(x, y)$  de variables de  $V$  de nature différente (l'une de décision appartenant à une composante  $c$ , l'autre d'environnement appartenant à une composante  $c'$ ), telle que  $c$  est ascendante de  $c'$  dans  $G$ ,  $x$  n'apparaît pas à droite de  $y$  dans  $Sov$  (ce qui implique qu'elle apparaisse à gauche ou qu'elle soit libre).

### 3.3.2. Hypothèses sous-jacentes

Cette définition repose sur certaines hypothèses qu'il peut être important d'expliquer : (1) l'ordre selon lequel les décisions sont prises par les différents agents et les observations sont faites est supposé connu ; (2) quand il prend une décision, un agent est informé de toutes les observations précédentes ; (3) un agent a un comportement optimiseur (maximiseur ou minimiseur selon son caractère coopératif ou non) ; (4) de plus, tous les agents cherchent à maximiser ou à minimiser la même quantité (la même utilité espérée).

De plus, nous supposons maintenant que l'ordre  $\preceq_u$  sur les utilités est total.

### 3.3.3. Valeur sémantique

Il est possible de définir la valeur d'une requête en utilisant les concepts d'*arbre de décision* et de *loterie* présentés dans la section 2.3. Le résultat est le suivant sous l'hypothèse que la structure de plausibilité soit conditionnable (voir la section 3.2.5) et que toutes les variables soient liées (si toutes les variables ne sont pas liées, la valeur d'une requête est simplement une fonction de l'affectation des variables libres) :

**Définition 7** La valeur sémantique  $\mathcal{V}_S(Sov, P)$  d'une requête correcte  $Sov$  sur un PFU  $P$  est récursivement définie de la façon suivante, si  $\mathcal{P}_V$ ,  $\mathcal{F}_V$  et  $\mathcal{U}_V$  sont respectivement les distributions jointes de plausibilité, de faisabilité et d'utilité associées à  $P$  :

- (1)  $\mathcal{V}_S(Sov, P) = \mathcal{V}_{Sr}(Sov, P, \emptyset)$
- (2)  $\mathcal{V}_{Sr}(\emptyset, P, A) = \mathcal{U}_V(A)$
- (3)  $(x \in V_D) \wedge (op = \min) \rightarrow$   
 $\mathcal{V}_{Sr}((op, x).Sov, P, A) = \min_{a \in Dom(x)} \mathcal{F}_V((x, a)|A) \star \mathcal{V}_{Sr}(Sov, P, A.(x, a))$
- (4)  $(x \in V_D) \wedge (op = \max) \rightarrow$   
 $\mathcal{V}_{Sr}((op, x).Sov, P, A) = \max_{a \in Dom(x)} \mathcal{F}_V((x, a)|A) \star \mathcal{V}_{Sr}(Sov, P, A.(x, a))$
- (5)  $(x \in V_E) \wedge (op = \oplus_u) \rightarrow$   
 $\mathcal{V}_{Sr}((op, x).Sov, P, A) = \oplus_u \mathcal{P}_V((x, a)|A) \otimes_{pu} \mathcal{V}_{Sr}(Sov, P, A.(x, a))$

Cette définition, sémantiquement justifiée par les concepts d'arbre de décision et de loterie, pose des problèmes au niveau algorithmique. En effet, le calcul de  $\mathcal{V}_S(Sov, P)$  nécessite un appel répété au calcul de quantités telles que  $\mathcal{F}_V((x, a)|A)$  et  $\mathcal{P}_V((x, a)|A)$  qui ne sont pas présentes telles quelles dans la définition du PFU et dont le calcul (de type calcul d'une probabilité marginale) peut être de complexité exponentielle en fonction du nombre de variables non affectées par  $A$ .

### 3.3.4. Valeur opérationnelle

Il est cependant possible de considérer une autre définition de la valeur d'une requête qu'on peut qualifier d'*opérationnelle* dans la mesure où elle ne présente pas les mêmes difficultés algorithmiques que la précédente. En supposant que toutes les variables sont liées (l'hypothèse d'une structure de plausibilité conditionnable n'est pas ici nécessaire) :

**Définition 8** *La valeur opérationnelle  $\mathcal{V}_O(Sov, P)$  d'une requête correcte  $Sov$  sur un PFU  $P$  est récursivement définie de la façon suivante :*

- (1)  $\mathcal{V}_O(Sov, P) = \mathcal{V}_{Or}(Sov, P, \emptyset)$
- (2)  $\mathcal{V}_{Or}(\emptyset, P, A) = ((\bigwedge_{F_i \in F} F_i) \star ((\otimes_{P_i \in P} P_i) \otimes_{pu} (\otimes_{U_i \in U} U_i)))(A)$
- (3)  $\mathcal{V}_{Or}((op, x).Sov, P, A) = op_{a \in Dom(x)} \mathcal{V}_{Or}(Sov, P, A.(x, a))$

Dans la mesure où elle fait appel uniquement à des quantités présentes dans la définition du PFU, cette définition est algorithmiquement plus attrayante. Sa justification sémantique est cependant moins claire.

### 3.3.5. Théorème d'équivalence

Il est heureusement possible de lever ces difficultés en établissant le théorème suivant d'équivalence entre les deux définitions proposées :

**Théorème 2** *Si la structure de plausibilité est conditionnable, les valeurs sémantique et opérationnelle d'une requête correcte sont identiques :  $\mathcal{V}_S(Sov, P) = \mathcal{V}_O(Sov, P)$ . De plus, les règles de décision qui peuvent être obtenues dans les deux cas en mémorisant pour chaque décision la(les) valeur(s) optimisante(s) (maximisante(s) ou minimisante(s), l'argmax ou l'argmin) sont identiques.*

Ce théorème donne une justification sémantique à la définition opérationnelle algorithmiquement intéressante, ce qui ne veut pas dire qu'avec cette définition, les choses seront forcément faciles (voir la section 4).

### 3.3.6. Requêtes classiques dans les cadres couverts

Les requêtes classiques dans les cadres couverts par le cadre PFU (voir la section 3.2.6) sont couvertes par la structure générique proposée pour les requêtes.

Avec un *problème de satisfaction de contraintes* (CSP)  $P = (V, C)$ , la requête classique est  $Sov = \{(\max, V)\}$  (avec  $\max = \vee$ ). Il en est de même pour un *problème de satisfiabilité d'une formule booléenne* (SAT). Pour un CSP *valué*, elle est aussi  $Sov = \{(\max, V)\}$ , avec un  $\max$  qui dépend de la structure de valuation choisie.

Avec une *formule booléenne quantifiée* (QBF) ou un CSP *quantifié* (QCSP), apparaît une libre alternance entre  $\min = \wedge$  et  $\max = \vee$ . Avec un *problème de satisfiabilité stochastique* (SSAT) ou un CSP *stochastique* (SCSP), on trouve aussi une libre alternance entre  $\max$  pour les variables de décision et  $+$  pour les variables d'environnement.

Avec un *réseau bayésien*, le calcul d'une distribution de probabilité sur une variable  $v$  se ramène à des requêtes de type  $Sov = \{(+, V - \{v\})\}$ . La recherche de l'hypothèse la plus vraisemblable sur la valeur d'une variable  $v$  amène à considérer cette variable comme une variable de décision et à considérer une requête de type  $Sov = \{(\max, \{v\}), (+, V - \{v\})\}$ .

Avec un *diagramme d'influence* ou un *réseau de valuation*  $P$ , on a une alternance entre  $\max$  pour les variables de décision et  $+$  pour les variables d'environnement, sachant que l'ordre d'élimination doit être cohérent avec le DAG associé à  $P$ .

Avec un *processus décisionnel markovien* à horizon fini (MDP), on a une alternance, à chaque instant  $i$ , entre  $\max$  pour  $d_i$  et  $+$  pour  $s_i$ . Avec un *processus partiellement observable* (POMDP), on a la même alternance entre  $\max$  pour  $d_i$  et  $+$  pour  $o_i$ , mais tous les  $s_i$  (qui sont non directement observables) sont repoussés à la fin de la séquence et éliminés avec  $+$ .

Avec un *problème de planification* à horizon fixé dans le cadre STRIPS, on se retrouve dans la même situation que dans le cadre CSP, du fait que  $\max = \oplus_u = \vee$ .

#### 4. Des algorithmes génériques

Sur la base du cadre proposé et de la définition opérationnelle de la valeur d'une requête, il est possible de définir des algorithmes génériques capables de répondre à n'importe quelle requête correcte sur n'importe quel PFU.

##### 4.1. Recherche arborescente

Un premier algorithme possible est un algorithme de type *recherche arborescente* qui utilise comme ordre d'affectation des variables un ordre compatible avec la requête  $Sov$  : si deux variables  $x$  et  $y$  appartiennent à deux paires (opérateur d'élimination-ensemble de variables) différentes  $pov_x$  et  $pov_y$  et que  $pov_x$  est située avant  $pov_y$  (à gauche) dans  $Sov$ ,  $x$  est affecté avant  $y$  ; si elles appartiennent à une même paire, leur ordre d'affectation est indifférent (du point de vue du résultat, pas forcément du point de vue de l'efficacité algorithmique).

Cet algorithme collecte au niveau de chaque feuille  $f$  la combinaison des plausibilités, des faisabilités et des utilités pour l'affectation complète associée à  $f$ . Il synthétise au niveau de chaque nœud non feuille  $n$  les valeurs fournies par les nœuds fils de  $n$  en utilisant l'opérateur d'élimination associé à  $n$ . La figure 2 montre le pseudo-code d'un tel algorithme (avec toujours l'hypothèse que toutes les variables sont liées). La réponse à la requête  $Sov$  sur un PFU  $P$  est fournie par l'appel **RechArbPFU**( $P, Sov, \emptyset$ ).

```

RechArbPFU( $P, Sov, A$ )
début
  si  $Sov = \emptyset$  alors
    retourner  $\left( (\wedge_{F_i \in F} F_i) \star \left( \otimes_{P_i \in P} P_i \right) \otimes_{pu} \left( \otimes_{U_i \in U} U_i \right) \right) (A)$ 
  sinon
     $(op, S).Sov' \leftarrow Sov$ 
    choisir  $x \in S$ 
    si  $S = \{x\}$  alors  $Sov \leftarrow Sov'$  sinon  $Sov \leftarrow (op, S - \{x\}).Sov'$ 
     $dom \leftarrow Dom(x)$ 
     $res \leftarrow \diamond$ 
    tant que  $dom \neq \emptyset$  faire
      choisir  $a \in dom$ 
       $dom \leftarrow dom - \{a\}$ 
       $res \leftarrow op(res, \mathbf{RechArbPFU}(P, Sov, A.(x, a)))$ 
    retourner  $res$ 
fin

```

**Figure 2:** Un algorithme générique naïf de type recherche arborescente.

Cet algorithme naïf a une complexité *temporelle exponentielle* en fonction du nombre de variables, mais une complexité *spatiale polynomiale*.

Ceci nous fournit au moins une information intéressante sur la *classe de complexité* du problème de réponse à une requête correcte sur un PFU, problème noté AnswerPFU : sachant que le problème QBF (satisfiabilité d'une formule booléenne quantifiée) est *PSPACE-complet* et est un sous problème du problème AnswerPFU, ce dernier est au moins *PSPACE-dur* ; mais sachant qu'il est *PSPACE* (existence d'un algorithme de complexité spatiale polynomiale), il est forcément *PSPACE-complet*<sup>2</sup>.

Un tel algorithme n'utilise pas le fait que la fonction globale d'utilité est le résultat de l'agrégation de fonctions locales de faisabilité, de plausibilité et d'utilité. La prise en compte de ce fait conduit à plusieurs améliorations possibles. Tout d'abord, les *infaisabilités* (faisabilités égales à  $f$ , associé aux décisions impossibles), les *impossibilités* (plausibilités égales à  $0_p$ , associé aux événements impossibles) et les *innacceptabilités* (utilités égales à  $\perp_u$ , possible élément minimum de  $E_u$ , associé aux situations inacceptables) peuvent être utilisées pour couper l'arborescence dès qu'elles sont détectées. Cette détection peut se faire de différentes façons, en s'inspirant des techniques de type *backward-checking*, *forward-checking* ou renforcement d'*arc-cohérence* utilisées pour traiter des contraintes locales dures dans le cadre CSP (Dechter, 2003). Les techniques de renforcement de *cohérence locale*, étendues des contraintes dures aux contraintes souples (Cooper *et al.*, 2004), peuvent par ailleurs permettre des calculs de borne utilisables pour couper l'arborescence à la façon d'un *branch and bound* en cas de nœuds min ou max. Il est aussi possible de s'inspirer des techniques utilisées dans les *algorithmes de jeu*, comme l'algorithme *alpha-beta* (Knuth *et al.*, 1975; Ballard, 1983), pour traiter plus efficacement des problèmes où alternent des nœuds min et max.

#### 4.2. Élimination de variables

Un second algorithme possible est un algorithme de type *élimination de variables* qui utilise comme ordre d'élimination des variables un ordre inverse de celui utilisé par la recherche arborescente : si deux variables  $x$  et  $y$  appartiennent à deux paires (opérateur d'élimination-ensemble de variables) différentes  $pov_x$  et  $pov_y$  et que  $pov_x$  est située avant  $pov_y$  (à gauche) dans  $Sov$ ,  $y$  est éliminée avant  $x$  ; si elles appartiennent à une même paire, leur ordre d'élimination est indifférent.

Cet algorithme commence par combiner toutes les fonctions locales de plausibilité, de faisabilité et d'utilité en une seule fonction globale d'utilité. Il en élimine ensuite les variables les unes après les autres en utilisant à chaque fois l'opérateur d'élimination associé. La figure 3 montre le pseudo-code d'un tel algorithme.

---

2. En toute rigueur, le problème AnswerPFU n'est pas un problème de décision et il faudrait parler du problème de décision associé, par exemple, le problème de déterminer si la valeur de la réponse est supérieure ou égale à une valeur fixée.

```

ElimVarPFU( $P, Sov$ )
début
   $L_0 \leftarrow \left( (\wedge_{F_i \in F} F_i) \star \left( \otimes_{P_i \in P} P_i \right) \otimes_{pu} \left( \otimes_{U_i \in U} U_i \right) \right)$ 
  tant que  $Sov \neq \emptyset$  faire
     $(op, S).Sov' \leftarrow Sov$ 
    choisir  $x \in S$ 
    si  $S = \{x\}$  alors  $Sov \leftarrow Sov'$  sinon  $Sov \leftarrow (op, S - \{x\}).Sov'$ 
     $L_0 \leftarrow op_x L_0$ 
  retourner  $L_0$ 
fin

```

**Figure 3:** Un algorithme générique naïf de type élimination de variables.

Cet algorithme naïf est pire que le précédent puisqu'il a une complexité *temporelle* et *spatiale exponentielle*. De la même façon que le premier algorithme de recherche arborescente présenté dans la figure 2, cet algorithme n'exploite pas la *factorisation* de la fonction globale d'utilité en fonctions locales de plausibilité, de faisabilité et d'utilité. Sans propriétés algébriques supplémentaires, il est cependant impossible d'utiliser des algorithmes d'élimination classiques, comme par exemple *bucket elimination* (Dechter, 1999), qui exploitent la structure de l'hyper-graphe des fonctions locales. Nous avons retenu deux ensembles de propriétés supplémentaires, parce qu'ils sont l'un ou l'autre satisfaits par chacun des cadres classiques et parce qu'ils permettent l'un et l'autre l'utilisation de ces algorithmes d'élimination classiques.

Le premier ensemble de propriétés est défini par :

$$((E_p, \oplus_p, \otimes_p) = (E_u, \oplus_u, \otimes_u)) \wedge (\otimes_{pu} = \otimes_u) \wedge (\preceq_p = \preceq_u) \quad [9]$$

Il implique que plausibilités et utilités utilisent la même structure de *semi-anneau* commutatif  $(E, \oplus, \otimes)$ , équipé d'un ordre total  $\preceq$  dont  $0_E$  est l'élément minimum, avec monotonie de  $\oplus$  et  $\otimes$  vis-à-vis de  $\preceq$ <sup>3</sup>. Ces propriétés sont satisfaites par les cadres 2, 3, 5 et 6 du tableau 1.

Le second ensemble de propriétés est simplement défini par :

$$\oplus_u = \otimes_u \text{ (sur } E_u, \text{ pas sur } E_u \cup \{\diamond\}) \quad [10]$$

Il implique que les utilités soient agrégées et éliminées avec le même opérateur. Ces propriétés, qui s'approchent de celles d'un *espace vectoriel*, sont satisfaites par les cadres 1, 4 et 7 du tableau 1.

3. Pour que les faisabilités utilisent la même structure, nous les exprimons sur  $\{1_E, \diamond\}$  au lieu de  $\{t, f\}$ , ce qui préserve la valeur de la réponse à une requête.



La figure 4 montre un algorithme de type *bucket elimination* applicable dans le cas d'une structure de *semi-anneau*. La réponse à la requête  $Sov$  sur un PFU  $P$  est fournie par l'appel  $\mathbf{ElimVarPFU-SA}(\mathcal{L}, Sov)$ , où  $\mathcal{L}$  est l'ensemble indifférencié des fonctions locales de plausibilité, de faisabilité et d'utilité de  $P$ .

```

ElimVarPFU-SA( $\mathcal{L}, Sov$ )
début
  si  $Sov = \emptyset$  alors retourner  $\mathcal{L}$ 
  sinon
     $(op, S).Sov' \leftarrow Sov$ 
    choisir  $x \in S$ 
    si  $S = \{x\}$  alors  $Sov \leftarrow Sov'$  sinon  $Sov \leftarrow (op, S - \{x\}).Sov'$ 
     $\mathcal{L}^{+x} \leftarrow \{L_i \in \mathcal{L} \mid x \in S(L_i)\}$ 
     $L_0 \leftarrow op_x(\otimes_{L_i \in \mathcal{L}^{+x}} L_i)$ 
     $\mathcal{L} \leftarrow (\mathcal{L} - \mathcal{L}^{+x}) \cup \{L_0\}$ 
    retourner  $\mathbf{ElimVarPFU-SA}(\mathcal{L}, Sov)$ 
fin

```

**Figure 4:** Un algorithme générique de type bucket elimination dans le cas d'une structure de semi-anneau.

On notera qu'avec cet algorithme, les fonctions locales ne sont pas combinées en une seule fois. Elles sont combinées progressivement par blocs puisque, lors de l'élimination d'une variable  $x$ , seules sont combinées les fonctions locales faisant intervenir  $x$ . Ceci nous permet de réduire la complexité temporelle et spatiale de l'algorithme 3, qui n'est plus une fonction exponentielle du nombre  $n$  de variables, mais une fonction exponentielle de la *largeur d'arbre*  $w$  de l'hyper-graphe des fonctions locales, suivant tous les ordres compatibles avec la requête  $Sov$ , avec  $w \leq n$ .

En fait, on peut profiter dans ce cas de tous les travaux algorithmiques menés dans le cadre algébrique des semi-anneaux commutatifs (Bertelé *et al.*, 1972; Shenoy, 1991; Kohlas, 2003), en particulier de tous les algorithmes de type *programmation dynamique* sur des décompositions de l'hyper-graphe des fonctions locales en *arbres de clusters*. On peut aussi profiter de tous les algorithmes de *recherche arborescente* utilisant des mécanismes de *mémorisation* qui exploitent aussi des décompositions de l'hyper-graphe des fonctions locales en arbres de clusters (Darwiche, 2001; Jégou *et al.*, 2003) et évitent la résolution répétée de sous-problèmes identiques.

Dans le cas d'une structure de type *espace vectoriel*, travailler sur des paires (plausibilité, utilité), comme cela se fait pour traiter des diagrammes d'influence (Ndilikilikisha, 1994), est une option disponible pour se ramener à une structure de semi-anneau. Mais d'autres options existent, comme par exemple des mécanismes de décomposition de la requête multi-opérateur initiale en un DAG de requêtes élémentaires mono-opérateurs sur des structures de semi-anneau.

### 4.3. D'autres approches algorithmiques

Dans une autre direction, des algorithmes de *recherche locale* (Aarts *et al.*, 1997) ou d'*échantillonnage* (Doucet *et al.*, 2000) sont de bons candidats dans le cadre d'une *résolution approchée* : recherche locale pour traiter des éliminations de type min ou max, échantillonnage pour traiter des éliminations de type +.

## 5. Conclusion

Dans cet article, nous avons montré que les problèmes posés dans de nombreux formalismes utilisés en Intelligence Artificielle se ramènent à des séquences d'éliminations de variables sur des agrégations de fonctions locales et peuvent être capturés dans un formalisme générique, dénommé PFU, dont les composants sont (1) une structure algébrique de plausibilité, de faisabilité et d'utilité, (2) une structure graphique à base de variables, de composantes, de fonctions locales de plausibilité, de faisabilité et d'utilité, de DAG entre composantes et de normalisations des fonctions locales de plausibilité et de faisabilité au niveau de chaque composante et (3) une structure des requêtes sous la forme d'une séquence de paires (opérateur d'élimination-ensemble de variables).

Nous avons indiqué une démarche naturelle de définition d'un PFU basée sur les notions de composantes et d'indépendance conditionnelle entre composantes. Nous avons aussi indiqué une démarche naturelle de définition d'une requête sur la base de l'ordre entre observations et décisions et du comportement coopératif ou non des agents décideurs. Nous avons donné un sens à la réponse à une requête correcte, via le théorème d'équivalence entre valeurs sémantique et opérationnelle d'une requête.

Ce cadre générique est intéressant en soi, car il permet de comparer dans un même cadre des formalismes développés par des communautés différentes. On peut par exemple remarquer que la structure utilisée pour des MDP possibilistes optimistes à horizon fini (complètement ou partiellement observables) est identique à celle utilisée pour des CSP valués possibilistes et en déduire que les mêmes algorithmes peuvent être utilisés dans les deux cas. Il en est de même entre la structure utilisée pour des problèmes de planification à horizon fixé dans le cadre STRIPS et celle utilisée dans les cadres SAT et CSP. Ce cadre générique permet aussi de considérer de nouveaux formalismes en jouant simplement sur la structure utilisée. Considérer par exemple une version possibiliste des diagrammes d'influence est immédiat dans le cadre proposé.

Le travail restant porte essentiellement sur l'algorithmique générique qu'il est possible de développer au dessus du cadre proposé. Sur ce point, le cadre PFU se présente comme un lieu d'accueil, de comparaison et de combinaison pour toutes les idées algorithmiques développées dans les divers cadres qu'il inclue : algorithmes de recherche arborescente, coupes de type *branch and bound* ou *alpha beta*, méthodes de renforcement de cohérence locale sur des contraintes dures ou souples, algorithmes à base d'élimination de variables, algorithmes de programmation dynamique ou de recherche arborescente avec mémorisation sur des décompositions arborescentes de

l'hypergraphe des fonctions locales, algorithmes de recherche locale ou d'échantillonnage ...

## 6. Remerciements

Nous remercions Jérôme Lang de l'IRIT, Régis Sabbadin de l'INRA et Jean-Loup Farges de l'ONERA pour leurs encouragements, contributions et commentaires. Nous remercions aussi le LAAS et l'INRA qui ont apporté leur soutien financier, pour le LAAS dans le cadre du projet européen COGNIRON (*The Cognitive Companion*).

## 7. Bibliographie

- Aarts E., Lenstra J. (eds), *Local Search in Combinatorial Optimization*, John Wiley & Sons, 1997.
- Ballard B., « The \*-Minimax Search Procedure for Trees containing Chance Nodes », *Artificial Intelligence*, vol. 21, n° 3, p. 327-350, 1983.
- Bertelé U., Brioschi F., *Nonserial Dynamic Programming*, Academic Press, 1972.
- Bistarelli S., Montanari U., Rossi F., Schiex T., Verfaillie G., Fargier H., « Semiring-Based CSPs and Valued CSPs: Frameworks, Properties and Comparison », *Constraints*, vol. 4, n° 3, p. 199-240, 1999.
- Bordeaux L., Montfroy E., « Beyond NP: Arc-consistency for Quantified Constraints », *Proc. of the 8th International Conference on Principles and Practice of Constraint Programming (CP-02)*, Ithaca, New York, USA, p. 371-386, 2002.
- Boutilier C., Brafman R., Hoos H., Poole D., « Reasoning with Conditional Ceteris Paribus Preference Statements », *Proc. of the 15th International Conference on Uncertainty in Artificial Intelligence (UAI-99)*, Stockholm, Sweden, 1999.
- Boutilier C., Dearden R., Goldszmidt M., « Stochastic Dynamic Programming with Factored Representations », *Artificial Intelligence*, vol. 121, n° 1-2, p. 49-107, 2000.
- Chu F., Halpern J., « Great Expectations. Part I: On the Customizability of Generalized Expected Utility », *Proc. of the 18th International Joint Conference on Artificial Intelligence (IJCAI-03)*, Acapulco, Mexico, 2003.
- Cooper M., Schiex T., « Arc Consistency for Soft Constraints », *Artificial Intelligence*, vol. 154, n° 1-2, p. 199-227, 2004.
- Darwiche A., « Recursive Conditioning », *Artificial Intelligence*, vol. 126, p. 5-41, 2001.
- Dechter R., « Bucket Elimination: a Unifying Framework for Reasoning », *Artificial Intelligence*, vol. 113, p. 41-85, 1999.
- Dechter R., *Constraint Processing*, Morgan Kaufmann, 2003.
- Dechter R., Larkin D., « Hybrid Processing of Beliefs and Constraints », *Proc. of the 17th International Conference on Uncertainty in Artificial Intelligence (UAI-01)*, Seattle, WA, USA, p. 112-119, 2001.
- Doucet A., de Freitas N., Murphy K., Russell S., « Rao-Blackwellised Particle Filtering for Dynamic Bayesian Networks », *Proc. of the 16th International Conference on Uncertainty in Artificial Intelligence (UAI-00)*, Stanford, CA, USA, p. 176-183, 2000.

- Dubois D., Prade H., *Possibility Theory: An Approach to Computerized Processing of Uncertainty*, Plenum Press, 1988.
- Fargier H., Lang J., Martin-Clouaire R., Schiex T., « A Constraint Satisfaction Framework for Decision under Uncertainty », *Proc. of the 11th International Conference on Uncertainty in Artificial Intelligence (UAI-95)*, Montréal, Canada, 1995.
- Fargier H., Lang J., Schiex T., « Mixed Constraint Satisfaction: a Framework for Decision Problems under Incomplete Knowledge », *Proc. of the 13th National Conference on Artificial Intelligence (AAAI-96)*, Portland, OR, USA, p. 175-180, 1996.
- Fargier H., Perny P., « Qualitative Models for Decision under Uncertainty without the Commensurability Assumption », *Proc. of the 15th International Conference on Uncertainty in Artificial Intelligence (UAI-99)*, Stockholm, Sweden, 1999.
- Fikes R., Nilsson N., « STRIPS: a New Approach to the Application of Theorem Proving », *Artificial Intelligence*, vol. 2, p. 189-208, 1971.
- Friedman N., Halpern J., « Plausibility Measures: A User's Guide », *Proc. of the 11th International Conference on Uncertainty in Artificial Intelligence (UAI-95)*, Montréal, Canada, p. 175-184, 1995.
- Halpern J., « Conditional Plausibility Measures and Bayesian Networks », *Journal of Artificial Intelligence Research*, vol. 14, p. 359-389, 2001.
- Howard R., Matheson J., « Influence Diagrams », *Readings on the Principles and Applications of Decision Analysis*, Strategic Decisions Group, Menlo Park, CA, USA, p. 721-762, 1984.
- Jégou P., Terrioux C., « Hybrid Backtracking bounded by Tree-decomposition of Constraint Networks », *Artificial Intelligence*, vol. 146, n° 1, p. 43-75, 2003.
- Knuth D., Moore R., « An Analysis of Alpha Beta Pruning », *Artificial Intelligence*, vol. 6, n° 4, p. 293-326, 1975.
- Kohlas J., *Information Algebras: Generic Structures for Inference*, Springer, 2003.
- Kushmerick N., Hanks S., Weld D., « An Algorithm for Probabilistic Planning », *Artificial Intelligence*, vol. 76, p. 239-286, 1995.
- Littman M., Majercik S., Pitassi T., « Stochastic Boolean Satisfiability », *Journal of Automated Reasoning*, vol. 27, n° 3, p. 251-296, 2001.
- Mackworth A., « Consistency in Networks of Relations », *Artificial Intelligence*, vol. 8, n° 1, p. 99-118, 1977.
- McDermott D., « The 1998 AI Planning Systems Competition », *AI Magazine*, 2000.
- Monahan G., « A Survey of Partially Observable Markov Decision Processes: Theory, Models, and Algorithms », *Management Science*, vol. 28, n° 1, p. 1-16, 1982.
- Murphy K., *Dynamic Bayesian Networks: Representation, Inference, and Learning*, Phd thesis, UC Berkeley, CA, USA, 2002.
- Ndilikilikisha P., « Potential Influence Diagrams », *Journal of Approximated Reasoning*, vol. 10, p. 251-285, 1994.
- Pearl J., *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann, 1988.
- Perny P., Spanjaard O., Weng P., « Algebraic Markov Decision Processes », *Proc. of the 19th International Joint Conference on Artificial Intelligence (IJCAI-05)*, Edinburgh, Scotland, p. 1372-1377, 2005.

- Pralet C., Verfaillie G., Schiex T., An Algebraic Graphical Model for Decision with Uncertainties, Feasibilities, and Utilities, Technical report, LAAS-CNRS, 2005. <http://www.laas.fr/~cpralet/praletverfschiex.ps>.
- Shafer G., *A Mathematical Theory of Evidence*, Princeton University Press, 1976.
- Shenoy P., « Valuation-based Systems for Discrete Optimization », *Uncertainty in Artificial Intelligence*, vol. 6, p. 385-400, 1991.
- Shenoy P., « Valuation-based Systems for Bayesian Decision Analysis », *Operations Research*, vol. 49, n° 3, p. 463-484, 1992.
- Smith D., Weld D., « Conformant Graphplan », *Proc. of the 15th National Conference on Artificial Intelligence (AAAI-98)*, Madison, WI, USA, p. 889-896, 1998.
- Spohn W., « A General Non-probabilistic Theory of Inductive Reasoning », *Proc. of the 6th International Conference on Uncertainty in Artificial Intelligence (UAI-90)*, Cambridge, MA, USA, 1990.
- von Neumann J., Morgensten O., *Theory of Games and Economic Behaviour*, Princeton University Press, 1944.
- Walsh T., « Stochastic Constraint Programming », *Proc. of the 15th European Conference on Artificial Intelligence (ECAI-02)*, Lyon, France, 2002.