Algorithmes et complexités génériques pour différents cadres de décision séquentielle dans l'incertain

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

* *ONERA*
2, av. Édouard Belin, BP 4025, F-31055 Toulouse Cedex 4
cedric.pralet@onera.fr,gerard.verfaillie@onera.fr

** *INRA*
Chemin de Borde-Rouge, BP 52627, F-31326 Castanet-Tolosan Cedex
thomas.schiex@toulouse.inra.fr

RÉSUMÉ. *De nombreux cadres existent pour la représentation et la résolution de problèmes de décision séquentielle dans l'incertain. Récemment, un formalisme unifiant une partie importante de ces cadres, qu'ils soient probabilistes ou possibilistes, a été proposé sous le nom de formalisme PFU (pour Plausibilité, Faisabilité, Utilité). Cet article introduit des algorithmes génériques définis dans le cadre PFU et qui conséquemment sont directement applicables à tous les cadres couverts. Il est montré qu'une approche unifiée peut être définie quelle que soit la représentation de l'incertain utilisée. La complexité théorique des algorithmes introduits est également étudiée en fonction d'un paramètre appelé largeur induite contrainte.*

ABSTRACT. *Recently, a generic algebraic framework called PFU (for Plausibilities-Feasibilities-Utilities) has been introduced in order to represent generic forms of sequential decision making under uncertainties. The PFU framework covers various existing probabilistic or possibilistic formalisms. This article introduces generic algorithms defined in the PFU framework which are therefore applicable for various decision models. Theoretical complexity results are also provided based on a parameter called the constrained induced-width.*

MOTS-CLÉS : *cadre PFU, algorithmes génériques, élimination de variable, largeur induite.*

KEYWORDS: *PFU framework, generic algorithms, variable elimination, induced-width.*

DOI:10.3166/RIA.21.459-488

1. Motivations

Récemment, un cadre générique a été introduit pour représenter des problèmes de décision incluant éventuellement :

– des incertitudes sur l'état de l'environnement, exprimées sous forme de probabilités, de possibilités, de kappa-rankings... Ces incertitudes sont désignées par le terme général de *plausibilités* (Friedman *et al.*, 1995; Halpern, 2001) ;

– des contraintes sur les décisions, appelées des *faisabilités* ;

– des préférences sur les décisions et/ou sur l'environnement, appelées des *utilités* ;

– un aspect *séquentiel* lié à une alternance entre décisions d'agents différents ou à une alternance entre étapes de décision et étapes d'observation.

Le cadre ainsi développé, baptisé cadre PFU pour "Plausibilité-Faisabilité-Utilité" (Pralet *et al.*, 2006c; Pralet, 2006; Pralet *et al.*, 2007), permet de représenter des formes générales de problèmes de décision séquentielle avec plausibilités, faisabilités et utilités. Il généralise des cadres tels que les diagrammes d'influence probabilistes (Howard *et al.*, 1984) ou possibilistes (Garcia *et al.*, 2006), les processus décisionnels markoviens à horizon fini utilisant des probabilités (Puterman, 1994; Monahan, 1982), des possibilités (Sabbadin, 1999) ou des kappa-rankings (Spohn, 1990; Wilson, 1995), factorisés ou non (Boutilier *et al.*, 2000), les formules booléennes ou les problèmes de satisfaction de contraintes quantifiés (Bordeaux *et al.*, 2002), les problèmes dits de "conformant planning" (Goldman *et al.*, 1996)... De par sa flexibilité, le cadre PFU englobe aussi bien des approches probabilistes que des approches possibilistes.

Un des objectifs de cet article est de montrer que l'effort d'unification réalisé par le cadre PFU en termes de représentation de la connaissance peut être prolongé par un effort d'unification en termes algorithmiques. Cette unification algorithmique est possible car les algorithmes utilisables pour un formalisme donné dépendent essentiellement des propriétés algébriques des opérateurs utilisés pour combiner et synthétiser des informations (associativité, commutativité, distributivité...) et non de la nature probabiliste ou non de ce formalisme. L'approche définie occulte ainsi la sémantique des informations manipulées pour se concentrer sur des aspects purement algébriques. La complexité théorique des algorithmes introduits est également étudiée en fonction d'un paramètre appelé *largeur induite contrainte* qui dépend fortement de la structure du problème considéré. Ceci nous permet d'obtenir à moindre coût des résultats de complexité sur tous les formalismes couverts par le cadre PFU et de mettre à jour certaines classes polynomiales.

Dans son esprit, notre démarche est proche des travaux de définition d'algorithmes de programmation dynamique génériques (Bertelé *et al.*, 1972; Shenoy, 1990; Dechter, 1999; Aji *et al.*, 2000; Kolhas, 2003). La différence tient à la nature composite du cadre PFU, qui permet de combiner et de synthétiser les informations en utilisant divers opérateurs suivant que l'on manipule des plausibilités, des faisabilités ou des utilités et suivant que l'on cherche des décisions optimales ou des utilités espérées (cet

aspect multi-opérateur n'est pas présent dans les approches génériques classiques). L'intérêt d'une approche générique est de pouvoir (1) clairement comprendre les propriétés élémentaires permettant d'utiliser tel ou tel algorithme, (2) établir des liens entre divers algorithmes existants et (3) grâce à l'effort d'abstraction, faire bénéficier un cadre des avancées algorithmiques réalisées dans un autre cadre.

L'article est organisé de la manière suivante : nous rappelons tout d'abord succintement la définition du cadre PFU (section 2), nous introduisons ensuite un algorithme générique d'élimination de variables permettant de résoudre des problèmes de décision formulés dans le cadre PFU (section 3), avant d'inférer, à partir de l'algorithme unifié précédemment défini, des résultats de complexité théorique donnés en fonction d'un paramètre appelé largeur induite contrainte (section 4). Les preuves sont omises par manque de place mais sont disponibles dans (Pralet, 2006). [1]

2. Le cadre PFU

Le cadre PFU (Plausibilité-Faisabilité-Utilité) permet de représenter de manière générique des problèmes de décision séquentielle faisant intervenir des plausibilités, des faisabilités et des utilités. D'un point de vue très général, ce cadre repose sur trois éléments clés :

1) Le premier élément clé est une *structure algébrique générique* qui spécifie un modèle de décision dans l'incertain. Ce modèle définit comment les plausibilités (incertitudes) et les préférences sont manipulées, en utilisant notamment une forme d'utilité espérée généralisée.

2) Le deuxième élément clé correspond à un *réseau* composé de variables sur lesquelles portent des *fonctions locales*. Le terme "locales" indique que chaque fonction φ peut porter sur un sous-ensemble des variables du réseau. Ce sous-ensemble est appelé la portée de la fonction locale et est noté $sc(\varphi)$ ("sc" comme "scope"). Les informations sont ainsi exprimées de manière factorisée.

3) Enfin, le troisième élément clé correspond à des *requêtes génériques* permettant de spécifier des scénarios décisionnels variés. Un scénario décisionnel définit notamment un ordre dans lequel les décisions sont prises et les observations sont réalisées. La réponse à une requête correspond à l'utilité espérée optimale du scénario considéré (éventuellement accompagnée d'une politique optimale pour les décisions).

Ces trois éléments clés (une structure algébrique, un réseau de fonctions locales et des requêtes) sont introduits plus formellement ci-après. Se référer à (Pralet *et al.*, 2007) pour une présentation plus détaillée du cadre PFU.

---

1. Le travail décrit dans cet article a été initié lorsque le premier auteur travaillait à l'INRA Toulouse et au LAAS-CNRS.

2.1. *Préliminaires*

Dans cet article, nous aurons besoin de combiner des fonctions locales, par exemple pour calculer une somme de coûts locaux, une conjonction de contraintes, ou encore un produit de distributions de probabilité. Pour représenter cela de manière générique, nous parlerons de combinaison de fonctions locales utilisant un certain *opérateur de combinaison* $\otimes$ (respectivement égal à $+$, $\wedge$ et $\times$ dans les exemples précédents). Nous aurons également besoin de synthétiser des informations pour calculer une utilité minimale ou maximale générée par une décision x, ou encore une marginalisation probabiliste ou possibiliste sur une variable x. Nous utiliserons pour ce faire un certain opérateur $\oplus$ (respectivement égal à $\min$, $\max$, $+$ et $\max$ dans les cas précédents) appelé *opérateur d'élimination de* x : étant donnée une fonction locale φ et une variable x ayant un domaine fini de valeurs noté $dom(x)$, l'élimination de x de φ utilisant l'opérateur d'élimination $\oplus$ est la fonction locale de portée $sc(\varphi) - \{x\}$, notée $\oplus_x \varphi$, et définie par $(\oplus_x \varphi)(A) = \oplus_{a \in dom(x)} \varphi(A.(x, a))$ pour toute affectation A de $sc(\varphi) - \{x\}$. L'élimination d'un ensemble de variables se définit de manière similaire.

2.2. *Premier élément du cadre PFU : une structure algébrique générique*

2.2.1. *Expression des plausibilités, des faisabilités et des utilités*

Afin de représenter des incertitudes sur l'environnement de manière générique, le cadre PFU utilise une structure algébrique inspirée des travaux de (Friedman *et al.*, 1995; Halpern, 2001) et appelée *structure de plausibilité*. Cette structure spécifie comment les incertitudes sont combinées (via un opérateur $\otimes_p$) et éliminées (via un opérateur $\oplus_p$ permettant d'effectuer des marginalisations). Les situations impossibles et certaines auront respectivement des degrés de plausibilité notés 0_p et 1_p.

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

– $(E_p, \oplus_p, \otimes_p)$ *est un semi-anneau commutatif, ce qui signifie que les opérateurs binaires* $\oplus_p$ *et* $\otimes_p$ *(*$E_p \times E_p \rightarrow E_p$*) sont associatifs et commutatifs et possèdent chacun un élément neutre, noté* 0_p *pour* $\oplus_p$ *et* 1_p *pour* $\otimes_p$*, que* 0_p *est absorbant pour* $\otimes_p$ *et que* $\otimes_p$ *est distributif par rapport à* $\oplus_p$ *;*

– E_p *est équipé d'un ordre partiel* $\preceq_p$ *dont* 0_p *est l'élément minimum et* $\oplus_p$ *et* $\otimes_p$ *sont monotones pour cet ordre.*

Cette définition couvre aussi bien des modèles probabilistes (avec $(E_p, \oplus_p, \otimes_p) = (\mathbb{R}^+, +, \times)$) que des modéles possibilistes (avec par exemple $(E_p, \oplus_p, \otimes_p) = ([0, 1], \max, \min)$).

Les faisabilités sur les décisions, qui correspondent à des contraintes dures, sont quant à elles exprimées par des booléens $b \in \{t, f\}$. Ces faisabilités sont combinées par un $\wedge$ logique pour représenter le fait que des décisions sont faisables si et seulement si toutes les contraintes sur les décisions sont satisfaites.

Les utilités (préférences) sont exprimées sur une structure appelée *structure d'utilité*, qui spécifie la manière dont les utilités sont combinées.

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

– $(E_u, \otimes_u)$ est un monoïde commutatif, ce qui signifie que l'opérateur binaire $\otimes_u$ ($E_u \times E_u \rightarrow E_u$) est associatif et commutatif et possède un élément neutre noté 1_u ;

– E_u est équipé d'un ordre partiel $\preceq_u$ et $\otimes_u$ est monotone pour cet ordre.

Cette définition couvre aussi bien des modèles de type satisfaction (avec $\otimes_u = \wedge$) que des modèles de type optimisation avec critère utilitariste (avec $\otimes_u = +$) ou égalitariste (avec $\otimes_u = \min$).

2.2.2. *Prise en compte simultanée des plausibilités, des faisabilités et des utilités*

Afin de prendre en compte simultanément des incertitudes et des utilités, le cadre PFU utilise une *structure d'utilité espérée*. Inspirée des travaux de (Chu *et al.*, 2003), auxquels certains axiomes sont ajoutés, cette structure permet d'utiliser une forme d'utilité espérée généralisée exprimant les définitions classiques telles que celle de l'utilité espérée probabiliste ($\sum_x (P_x \cdot U_x)$) ou celle de l'utilité espérée possibiliste optimiste ($\max_x \min(P_x, U_x)$) sous une forme algébrique générale du type $\oplus_{ux}(P_x \otimes_{pu} U_x)$, avec $\otimes_{pu}$ un opérateur permettant de combiner plausibilité et utilité et avec $\oplus_u$ un opérateur utilisé pour synthétiser une utilité espérée globale.

Définition 3. *Une structure d'utilité espérée est un tuple $(S_p, S_u, \oplus_u, \otimes_{pu})$ tel que :*

– $S_p = (E_p, \oplus_p, \otimes_p)$ est une structure de plausibilité,

– $S_u = (E_u, \otimes_u)$ est une structure d'utilité,

– $(E_u, \oplus_u, \otimes_{pu})$ est un semi-module sur S_p, ce qui signifie que $(E_u, \oplus_u)$ est un monoïde commutatif (voir la définition 2) avec un élément neutre noté 0_u et que l'opérateur binaire $\otimes_{pu}$ ($E_p \times E_u \rightarrow E_u$) satisfait 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 $\preceq_u$.

Le tableau 1 donne quelques exemples de structures d'utilité espérée. Les définitions adoptées permettent d'encapsuler des modèles de raisonnement aussi bien du type utilité espérée probabiliste (von Neumann *et al.*, 1944) que du type utilité espérée possibiliste optimiste ou pessimiste (Dubois *et al.*, 1995), ou encore du type utilité espérée booléenne ou basée sur des kappa-rankings (Giang *et al.*, 2000). Par contre, le cadre des fonctions de croyance (Shafer, 1976) ou la généralisation de l'utilité espérée possibiliste proposée dans (Giang *et al.*, 2005) ne sont pas couverts.

Enfin, les faisabilités sur les décisions sont prises en compte en utilisant un élément spécial noté $\Diamond$ et appelé *élément d'infaisabilité*, avec comme particularité

le fait qu'il soit absorbant pour tout opérateur de combinaison $\otimes$ (l'agrégation de quelque chose d'infaisable avec n'importe quoi est infaisable) et neutre pour tout opérateur d'élimination $\oplus$ (ce qui est infaisable est ignoré). Un opérateur spécial $\star : \{f, t\} \times (E_u \cup \{\Diamond\}) \rightarrow E_u \cup \{\Diamond\}$ appelé *opérateur de troncature*, est également introduit. Cet opérateur $\star$ est défini par $f \star e = \Diamond$ et $t \star e = e$ pour tout $e \in E_u \cup \{\Diamond\}$, de telle sorte que les décisions infaisables soient associées à la valeur $\Diamond$ ignorée par les opérateurs d'élimination. Le rôle de $\Diamond$ ne peut pas être joué par un élément de E_p ou E_u car $\Diamond$ doit notamment être ignoré par les opérateurs d'élimination $\min$ et $\max$.

	E_p	$\preceq_p$	$\oplus_p$	$\otimes_p$	E_u	$\preceq_u$	$\otimes_u$	$\oplus_u$	$\otimes_{pu}$
1	$\mathbb{R}^+$	$\leq$	$+$	$\times$	$\mathbb{R} \cup \{-\infty\}$	$\leq$	$+$	$+$	$\times$
2	$\mathbb{R}^+$	$\leq$	$+$	$\times$	$\mathbb{R}^+$	$\leq$	$\times$	$+$	$\times$
3	$[0, 1]$	$\leq$	$\max$	$\min$	$[0, 1]$	$\leq$	$\min$	$\max$	$\min$
4	$[0, 1]$	$\leq$	$\max$	$\min$	$[0, 1]$	$\leq$	$\min$	$\min$	$\max(1–p, u)$
5	$\mathbb{N} \cup \{\infty\}$	$\geq$	$\min$	$+$	$\mathbb{N} \cup \{\infty\}$	$\geq$	$+$	$\min$	$+$
6	$\{t, f\}$	$\prec_{bool}$	$\vee$	$\wedge$	$\{t, f\}$	$\preceq_{bool}$	$\wedge$	$\vee$	$\wedge$
7	$\{t, f\}$	$\preceq_{bool}$	$\vee$	$\wedge$	$\{t, f\}$	$\preceq_{bool}$	$\wedge$	$\wedge$	$\rightarrow$
8	$\{t, f\}$	$\preceq_{bool}$	$\vee$	$\wedge$	$\{t, f\}$	$\preceq_{bool}$	$\vee$	$\vee$	$\wedge$
9	$\{t, f\}$	$\preceq_{bool}$	$\vee$	$\wedge$	$\{t, f\}$	$\preceq_{bool}$	$\vee$	$\wedge$	$\rightarrow$

Tableau 1. *Ensembles et opérateurs utilisés dans plusieurs cadres classiques : (1) utilité espérée probabiliste avec utilités additives (permet de calculer l'espérance d'un gain ou d'un coût) ; (2) utilité espérée probabiliste avec utilités multiplicatives, aussi appelée satisfaction espérée probabiliste (permet de calculer la probabilité que des contraintes soient satisfaites) ; (3) utilité espérée possibiliste optimiste ; (4) utilité espérée possibiliste pessimiste ; (5) utilité qualitative avec kappa-rankings et utilités uniquement positives ; (6) utilité espérée booléenne optimiste avec utilités conjonctives (permet de savoir s'il existe un monde possible dans lequel tous les buts d'un ensemble de buts sont satisfaits), avec $\preceq_{bool}$ l'ordre tel que $f \prec_{bool} t$; (7) utilité espérée booléenne pessimiste avec utilités conjonctives (permet de savoir si dans tous les mondes possibles tous les buts d'un ensemble de buts sont satisfaits) ; (8) utilité espérée booléenne optimiste avec utilités disjonctives (permet de savoir s'il existe un monde possible dans lequel au moins un but d'un ensemble de buts est satisfait) ; (9) utilité espérée booléenne pessimiste avec utilités disjonctives (permet de savoir si dans tous les mondes possibles au moins un but d'un ensemble de buts est satisfait)*

La structure algébrique du cadre PFU s'appuie ainsi sur des structures algébriques classiques de monoïde, de semi-anneau et de semi-module, sur lesquelles certaines hypothèses de monotonie sont ajoutées. Elle permet de considérer de manière unifiée divers cadres de décision dans l'incertain.

2.3. *Deuxième élément du cadre PFU : une forme générale de modèle graphique*

A partir de la structure algébrique de raisonnement précédemment définie, il est possible d'exprimer de la connaissance sous la forme d'un réseau de variables sur

lesquelles portent des fonctions locales. Ce réseau est appelé un *réseau PFU*. Les variables d'un réseau PFU sont associées ou bien à des décisions, ou bien à des paramètres décrivant l'état de l'environnement. Les premières, qui sont contrôlables, sont appelées *variables de décision* et les secondes, qui sont incontrôlables, sont appelées *variables d'environnement*. Les fonctions locales peuvent quant à elles correspondre à plusieurs types d'information : les fonctions dites de *plausibilité* correspondent aux facteurs d'une distribution d'incertitude globale (par exemple à des distributions conditionnelles locales) ; les fonctions dites de *faisabilité* correspondent aux facteurs d'une distribution de faisabilité globale (par exemple à des contraintes locales sur les décisions) ; enfin, les fonctions dites d'*utilité* correspondent à des facteurs définissant un degré d'utilité global (ces facteurs peuvent par exemple être des gains ou des coûts locaux).

L'utilisation de fonctions locales permettant d'exprimer des quantités globales est notamment justifiée par des considérations liées à l'*indépendance conditionnelle* entre variables. Cette notion se retrouve implicitement au niveau du cadre PFU puisque les fonctions de plausibilité et de faisabilité manipulées doivent satisfaire certaines conditions de normalisation définies par un DAG (Directed Acyclic Graph). Ce DAG n'est pas défini sur des variables comme dans les réseaux bayésiens (Pearl, 1988) mais, de manière plus générale et comme dans les *chain graphs* (Frydenberg, 1990), sur des ensembles de variables appelés des *composantes*. Ceci entraîne que chaque fonction de plausibilité prise indépendamment n'est pas nécessairement une distribution conditionnelle normalisée. Plus précisément, pour toute composante c du DAG, nous aurons un ensemble de fonctions de plausibilité noté $Fact(c)$ dont la combinaison représentera la distribution conditionnelle sur c sachant ses parents $pa(c)$ dans le DAG, c'est-à-dire tel que $\mathcal{P}_{c\,|\,pa(c)} = \otimes_{p\,P_i \in Fact(c)} P_i$. Du fait de la condition de normalisation $\oplus_{p_c} \mathcal{P}_{c\,|\,pa(c)} = 1_p$, nous aurons $\oplus_{p_c}(\otimes_{p\,P_i \in Fact(c)} P_i) = 1_p$. Des remarques similaires s'appliquent aux fonctions de faisabilité.

Exemple. Considérons deux capteurs A et B dont on sait qu'au moins un des deux est en panne. Il est possible de faire un test fiable sur un des capteurs pour savoir s'il fonctionne. Si un test a été réalisé, alors on peut choisir un capteur à réparer. Si aucun test n'a été effectué, on s'interdit toute réparation. L'état final du système correspond à l'état de fonctionnement du couple de capteurs après réparation éventuelle.

Variables. Pour modéliser ce problème, nous définissons tout d'abord des variables : deux variables d'environnement ec_A et ec_B décrivant l'état initial des capteurs (valeur 0 si le capteur considéré est en panne, 1 sinon), une variable de décision te décrivant le capteur testé (valeur A, B, ou n si on ne fait pas de test), une variable d'environnement rte décrivant le résultat du test (valeur 0 si on trouve que le capteur est en panne, 1 si on trouve qu'il fonctionne, ou n si on ne fait pas de test), une variable de décision rp décrivant le capteur réparé (valeur A, B, ou n si on ne fait pas de réparation) et une variable d'environnement ecs décrivant l'état final du couple de capteurs (valeur 1 si les deux capteurs fonctionnent, 0 sinon).

Fonctions locales. Pour modéliser les incertitudes et les effets des décisions sur l'environnement, nous utilisons des fonctions de plausibilité :

– Deux fonctions de plausibilité décrivant l'état initial : une fonction $P_{ec_A,ec_B} : (ec_A = 0) \vee (ec_B = 0)$ qui porte sur les variables ec_A et ec_B et qui spécifie qu'au moins un capteur est en panne et une fonction $P_\emptyset$ qui assure que la disjonction de toutes les situations possibles est certaine, i.e. $\oplus_{p_{ec_A,ec_B}}(P_{ec_A,ec_B} \otimes_p P_\emptyset) = 1_p$ (avec des probabilités, $P_\emptyset = 1/3$; avec des possibilités, $P_\emptyset = 1$).

– Trois fonctions de plausibilité $P_{rte,ec_A,te} : (te = A) \rightarrow (rte = ec_A)$, $P_{rte,ec_B,te} : (te = B) \rightarrow (rte = ec_B)$ et $P_{rte,te} : (te = n) \rightarrow (rte = n)$ qui décrivent qu'un test donne l'état d'un capteur de manière fiable. Ces fonctions satisfont la condition de normalisation $\oplus_{p_{rte}}(P_{rte,ec_A,te} \otimes_p P_{rte,ec_B,te} \otimes_p P_{rte,te}) = 1_p$.

– Une fonction $P_{ecs,ec_A,ec_B,rp}$ qui spécifie la plausibilité que le couple de capteurs fonctionne sachant l'état initial de chacun des capteurs et le capteur réparé (nous avons $\oplus_{p_{ecs}} P_{ecs,ec_A,ec_B,rp} = 1_p$). Suivant la structure algébrique utilisée, $P_{ecs,ec_A,ec_B,rp}$ pourra être une probabilité conditionnelle exprimant qu'une réparation a une certaine probabilité d'échouer, une possibilité conditionnelle, une incertitude booléenne spécifiant toutes les situations possibles...

Pour modéliser la règle selon laquelle on ne se lance pas dans une réparation si aucun test n'a été effectué, nous introduisons la fonction de faisabilité $F_{te,rp} : (te = n) \rightarrow (rp = n)$. Enfin, pour modéliser d'éventuelles préférences, nous pouvons utiliser des fonctions d'utilité. Par exemple, deux fonctions d'utilité U_{te} et U_{rp} peuvent décrire que faire un test ou une réparation a un coût si on travaille avec des utilités additives, une fonction $U_{rp,ecs}$ peut spécifier le fait que l'on ne souhaite pas faire une réparation et obtenir un couple de capteurs défectueux. Le réseau PFU associé au problème des capteurs est donné à la figure 1.

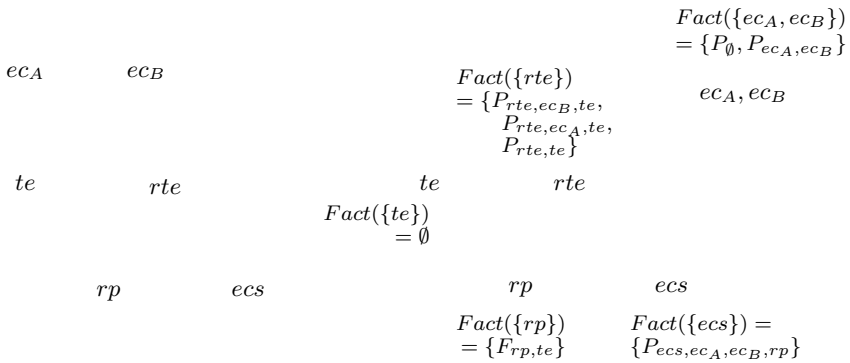


Figure 1. *Exemple de réseau PFU, avec à gauche le réseau de fonctions locales (la fonction locale $P_\emptyset$, dont la portée est vide, n'est pas représentée) et à droite le DAG représentant des conditions de normalisation sur les composantes. Les carrés sont associés aux variables de décision, les cercles aux variables d'environnement et les arêtes en trait continu (resp. double trait continu, pointillés) aux portées des fonctions locales de plausibilité (resp. faisabilité, utilité)*

Formellement, un réseau PFU est défini de la manière suivante :

Définition 4. *Un* réseau PFU *sur une structure d'utilité espérée est un quintuplet* (V, G, P, F, U) *tel que*

– $V = \{x_1, x_2, \ldots\}$ est un ensemble fini de variables à domaines finis. V est partitionné en un ensemble V_D de variables de décision et un ensemble V_E de variables d'environnement.

– G est un DAG dont les sommets, appelés composantes, sont une partition de V, plus précisément l'union d'une partition $\mathcal{C}_D$ de V_D et d'une partition $\mathcal{C}_E$ de V_E.

– $P = \{P_1, P_2, \ldots\}$ est un ensemble fini de fonctions de plausibilité. Chaque $P_i \in P$ est associée à une unique composante $c \in \mathcal{C}_E$ qui satisfait $sc(P_i) \subset c \cup pa_G(c)$. L'ensemble des fonctions de plausibilité $P_i \in P$ associées à une composante $c \in \mathcal{C}_E$ est noté $Fact(c)$ et doit satisfaire $\oplus_{p_c}(\otimes_{p\, P_i \in Fact(c)} P_i) = 1_p$.

– $F = \{F_1, F_2, \ldots\}$ est un ensemble fini de fonctions de faisabilité. Chaque $F_i \in F$ est associée à une unique composante $c \in \mathcal{C}_D$ qui satisfait $sc(F_i) \subset c \cup pa_G(c)$. L'ensemble des fonctions de faisabilité $F_i \in F$ associées à une composante $c \in \mathcal{C}_D$ est noté $Fact(c)$ et doit satisfaire $\vee_c(\wedge_{F_i \in Fact(c)} F_i) = t$.

– $U = \{U_1, U_2, \ldots\}$ est un ensemble fini de fonctions d'utilité.

Nous faisons ici deux hypothèses supplémentaires sur les réseaux PFU considérés :

– les variables d'une même composante sont d'une certaine façon liées indépendamment de leurs parents (sinon elles ne doivent pas être dans la même composante). Formellement, cela signifie que pour toute composante c, l'hypergraphe représentant les portées des fonctions locales de $Fact(c)$ restreintes à c doit être connexe ;

– une composante doit être d'une certaine manière liée à ses parents. Formellement, cela signifie que d'une part pour toute composante c et pour toute composante c' parente de c, il existe une fonction locale de $Fact(c)$ dont la portée contient une variable de c' et une variable de c, et que d'autre part une variable de c' qui intervient dans $Fact(c)$ apparaît dans au moins une fonction de $Fact(c)$ impliquant une variable de c.

Ces deux hypothèses sont relativement naturelles et il est possible de montrer que la première ne nuit pas à la généralité du cadre et que la seconde peut être faite dès que la structure de plausibilité satisfait $\forall p_1, p_2, p,\ (p_1 \otimes_p p = p_2 \otimes_p p = 1_p) \rightarrow (p_1 = p_2)$ (cette condition assez faible est satisfaite par toutes les structures de plausibilité classiques).

2.4. *Troisième élément du cadre PFU : des requêtes génériques*

Les problèmes de décision sur un réseau PFU sont formalisés sous la forme de *requêtes*. Les requêtes considérées font l'hypothèse qu'une séquence de décisions doit être prise (pas de décision parallèle), que l'ordre dans lequel des observations sont

réalisées et les décisions sont prises est complètement connu et enfin que l'ensemble des degrés d'utilité E_u est totalement ordonné par $\preceq_u$.

Une requête est modélisée par une séquence de couples opérateur-variable(s) notés (op, S). Un couple (op, S) représente implicitement l'élimination de l'ensemble de variables S avec l'opérateur op et est donc également noté par la suite sous la forme op_S. Nous donnons quelques exemples de requêtes avant d'introduire une définition formelle.

Exemple. Une requête définie par la séquence opérateur-variables $\max_{te} \oplus_{urte} \max_{rp} \oplus_{uecs,ec_A,ec_B}$ représentera le scénario "$\{te\}^+ \rightarrow \{rte\} \rightarrow \{rp\}^+ \rightarrow \{ecs, ec_A, ec_B\}$" dans lequel on choisit d'abord un meilleur test à effectuer, puis on observe le résultat, puis on choisit de faire la meilleure réparation possible, alors que les valeurs des variables ecs, ec_A et ec_B restent non observées avant la dernière décision.

Si on suppose que les deux décisions (test et réparation) sont prises par deux agents différents et que l'agent choisissant s'il faut réparer souhaite évaluer ce qui se passe si le test est choisi de la pire manière possible, alors la séquence opérateur-variables $\min_{te} \oplus_{urte} \max_{rp} \oplus_{uecs,ec_A,ec_B}$ pourra être utilisée. Pour modéliser un scénario dans lequel l'état du capteur A sera connu avant toute décision, nous utiliserons la séquence opérateur-variables $\oplus_{uec_A} \min_{te} \oplus_{urte} \max_{rp} \oplus_{uecs,ec_B}$. Ainsi, l'ordre des éliminations définit diverses situations observationnelles et l'opérateur d'élimination utilisé pour une décision définit si on considère que cette décision est prise de la pire ou de la meilleure manière possible.

Définition 5. *Une requête sur un réseau PFU est un couple $(Sov, \mathcal{N})$ tel que $\mathcal{N} = (V, G, P, F, U)$ est un réseau PFU et tel que Sov est une séquence de paires "(opérateur d'élimination op - ensemble de variables S)", aussi notées op_S. La séquence Sov doit être telle que*

1) tous les S sont disjoints ;

2) soit $S \subseteq V_D$ (variables de décision) et $op = \min$ ou $\max$, soit $S \subseteq V_E$ (variables d'environnement) et $op = \oplus_u$;

3) les variables qui n'apparaissent pas dans Sov sont des variables de décision ;

4) pour toute paire (x, y) de variables de V de natures différentes (l'une est une variable de décision, l'autre est une variable d'environnement) telle que la composante contenant x est ascendante de la composante contenant y dans G, x n'apparaît pas à droite de y dans Sov.

La dernière condition indique que le scénario décisionnel induit par Sov doit être compatible avec la causalité : elle permet d'exclure des requêtes dans lesquelles une observation serait réalisée avant la prise d'une décision influençant cette observation. Notons que rien n'interdit que dans certains cadres comme les cadres possibilistes, $\oplus_u = \min$ ou $\oplus_u = \max$, ce qui veut dire que $\min$ et $\max$ peuvent éventuellement être utilisés pour éliminer des variables d'environnement.

La définition d'une requête n'impose pas que toutes les variables d'un réseau PFU apparaissent dans Sov. Nous considérons par la suite sans perte de généralité que les requêtes considérées font apparaître toutes les variables de V dans Sov.

Réponse à une requête

Une requête exprime un problème de décision. La réponse à ce problème est définie de la manière suivante :

Définition 6. *La réponse $Ans(Q)$ à une requête $Q = (Sov, (V, G, P, F, U))$ est*

$$Ans(Q) = Sov((\underset{F_i \in F}{\wedge} F_i) \star (\underset{P_i \in P}{\otimes_p} P_i) \otimes_{pu} (\underset{U_i \in U}{\otimes_u} U_i)) \quad [1]$$

Nous supposons ici que le calcul de $Ans(Q)$ peut, si besoin est, s'accompagner d'un enregistrement des politiques optimales pour les décisions en utilisant des argmin et des argmax. Il est prouvé (Pralet, 2006) que cette définition, qui utilise uniquement les données brutes disponibles dans un réseau PFU, est équivalente à une définition à base d'arbre de décision dans le sens où les deux approches définissent non seulement la même utilité espérée optimale mais aussi les mêmes politiques optimales pour les décisions. Autrement dit, répondre à une requête en utilisant un arbre de décision représentatif du scénario décisionnel considéré est équivalent à calculer une séquence d'éliminations de variables sur la combinaison de toutes les fonctions locales du réseau PFU.

Les algorithmes définis dans le cadre PFU ont ainsi à calculer la valeur de l'équation 1. Etant donnée la forme de cette équation, il est assez naturel de se tourner pour la résolution vers un algorithme classique d'*élimination de variables*. Ce type d'algorithme, initialement appelé *programmation dynamique non sérielle* par Bertelé-Brioschi (Bertelé *et al.*, 1972) et Shenoy-Shafer (Shenoy *et al.*, 1988), peut également être rencontré dans la littérature sous des noms divers tels que *bucket elimination* (Dechter, 1999) ou encore *cluster-tree elimination* (Dechter, 2003). La section suivante va introduire un algorithme d'élimination de variables générique applicable aux formalismes couverts par le cadre PFU.

3. Un algorithme unifié d'élimination de variables

Les algorithmes classiques d'élimination de variables ont principalement été définis pour calculer des quantités génériques de la forme $\oplus_V (\otimes_{\varphi \in \Phi} \varphi)$, dans lesquelles V, Φ, $\oplus$, $\otimes$ sont respectivement un ensemble de variables, un ensemble de fonctions locales, un opérateur d'élimination et un opérateur de combinaison, avec $\oplus$ et $\otimes$ qui satisfont certaines propriétés algébriques élémentaires (commutativité, associativité, distributivité...). Le calcul $\oplus_V (\otimes_{\varphi \in \Phi} \varphi)$ correspond à un problème dit mono-opérateur car il n'implique qu'un seul opérateur d'élimination et qu'un seul opérateur de combinaison.

Les algorithmes d'élimination de variables ont été étendus pour traiter des problèmes multi-opérateurs impliquant plusieurs opérateurs d'élimination et/ou plusieurs opérateurs de combinaison, par exemple pour des problèmes de calcul de politique optimale pour un diagramme d'influence probabiliste (Ndilikilikesha, 1994) ou des problèmes de calcul d'explication optimale pour un réseau bayésien (Park *et al.*, 2003). Ces extensions à des problèmes multi-opérateurs restent néanmoins restreintes à des cadres spécifiques et n'ont pas la généralité algébrique des algorithmes d'élimination de variables pour les problèmes mono-opérateurs. L'ambition est ici de combler ce manque en définissant, à partir du cadre PFU, une forme générique (algébrique) d'algorithme d'élimination de variables multi-opérateur. Cet algorithme générique s'appliquera aussi bien à des cadres probabilistes qu'à des cadres possibilistes.

Nous rappelons brièvement le fonctionnement d'un algorithme d'élimination de variables mono-opérateur avant de présenter sa généralisation au cadre PFU.

3.1. *Algorithme classique d'élimination de variables mono-opérateur*

Supposons que l'on cherche à calculer la quantité suivante (un tel calcul peut être rencontré si on résout un VCSP additif (Schiex *et al.*, 1995)) :

$$\max_{x_1,x_2,x_3,x_4} (\varphi_{12} + \varphi_{13} + \varphi_{23} + \varphi_{14})$$

avec φ_{ij} qui désigne une fonction locale portant sur les variables x_i et x_j et à valeurs réelles. A chaque étape d'un algorithme classique d'élimination de variables, une variable x à éliminer est sélectionnée. L'élimination de x est alors réalisée en prenant en compte uniquement la partie du problème qui dépend de x. Par exemple, l'élimination de x_4 se fait en calculant $\varphi'_1 = \max_{x_4}(\varphi_{14})$, qui est une fonction unaire portant sur x_1. La quantité restant à calculer devient

$$\max_{x_1,x_2,x_3} (\varphi_{12} + \varphi_{13} + \varphi_{23} + \varphi'_1)$$

Si la variable suivante à éliminer est x_3, alors on calcule $\varphi'_{12} = \max_{x_3}(\varphi_{13} + \varphi_{23})$, qui est une fonction binaire portant sur x_1 et x_2. La quantité restant à évaluer devient

$$\max_{x_1,x_2}(\varphi_{12} + \varphi'_{12} + \varphi'_1)$$

Après élimination de toutes les variables, la valeur de l'expression initiale est obtenue. Il est également possible de mémoriser des argmax pour connaître une affectation optimale pour x_1, x_2, x_3, x_4.

En exploitant la factorisation d'une fonction globale en fonctions locales, ce type d'algorithme utilise la structure du problème considéré. Dans certaines approches telles que *cluster-tree elimination*, on s'autorise à éliminer à chaque étape non pas une seule variable mais un ensemble de variables, ce qui présente un intérêt en termes de taille mémoire utilisée. Mais le fonctionnement de l'algorithme est similaire et toutes

les approches introduites dans la suite de cet article, qui se concentrent sur des éliminations variable par variable, restent valides dans le cas d'éliminations d'ensembles de variables.

L'exemple introduit ci-dessus utilise un opérateur d'élimination $\oplus = \max$ et un opérateur de combinaison $\otimes = +$ définis sur un ensemble $E = \mathbb{R}$. De manière plus générale, un algorithme d'élimination de variables peut être utilisé dès que l'on considère un espace E et deux opérateurs $\oplus$ et $\otimes$ tels que $(E, \oplus, \otimes)$ est un semi-anneau commutatif. [2] L'algorithme générique correspondant, l'algorithme 1, est noté **VE** pour "**V**ariable **E**limination". Dans cet algorithme, étant donné un ensemble de fonctions locales Φ et une variable x, nous notons Φ^{+x} l'ensemble des fonctions ayant x dans leur portée ($\Phi^{+x} = \{\varphi \in \Phi \,|\, x \in sc(\varphi)\}$) et $\Phi^{-x} = \Phi - \Phi^{+x}$. A chaque étape, l'algorithme sélectionne une variable x à éliminer, effectue l'élimination de x sur la combinaison de toutes les fonctions locales qui dépendent de x et actualise l'ensemble des fonctions locales à considérer par la suite.

Les notations Φ^{+x} et Φ^{-x} utilisées dans l'algorithme **VE** sont également utilisées dans le reste de l'article.

Algorithme 1 : Un algorithme générique d'élimination de variables renvoyant $\oplus_V \left(\otimes_{\varphi \in \Phi} \varphi\right)$

VE(V, Φ)
début
  tant que $V \neq \emptyset$ **faire**
    choisir $x \in V$
    $V \leftarrow V - \{x\}$
    $\varphi_0 \leftarrow \oplus_x \left(\otimes_{\varphi \in \Phi^{+x}} \varphi\right)$
    $\Phi \leftarrow \Phi^{-x} \cup \{\varphi_0\}$
  retourner $\otimes_{\varphi \in \Phi} \varphi$
fin

3.2. *Adaptation au cadre PFU*

L'adaptation de l'algorithme précédent au cadre PFU soulève plusieurs problèmes, essentiellement liés à la nature composite et multi-opérateur du cadre PFU :

1) Un algorithme d'élimination de variables utilise intensément la propriété de distributivité de l'opérateur de combinaison $\otimes$ sur l'opérateur d'élimination $\oplus$. Cette propriété permet, lors de l'élimination d'une variable x, de ne considérer que les fonctions locales qui dépendent de x. Le premier problème est que les opérateurs de combinaison du cadre PFU ne sont pas nécessairement distributifs par rapport aux opérateurs

2. Des conditions encore plus faibles d'application d'un algorithme d'élimination de variables sont définies par le formalisme des algèbres de valuation (Kolhas, 2003).

d'élimination utilisés. C'est notamment le cas pour l'opérateur de combinaison des utilités $\otimes_u$ qui n'est pas supposé être distributif par rapport à $\oplus_u$.

Ceci engendre un problème appelé *problème de décomposabilité* : par exemple, une quantité telle que $\sum_x (P_{xy} \times \min(U_x, U_{zt}))$, [3] qui utilise un opérateur de combinaison $\otimes_u = \min$ qui n'est pas distributif par rapport à l'opérateur d'élimination $\oplus_u = +$ ne peut pas être décomposé d'une manière classique sous la forme $\min\left(\sum_x (P_{xy} \cdot U_x), U_{zt}\right)$ pour n'avoir à considérer que les fonctions locales qui dépendent de x pour éliminer x.

De manière plus générale, étant donnés des ensembles P, F et U de fonctions locales de plausibilité, de faisabilité et d'utilité respectivement, notons $(\wedge_{F_i \in F} F_i) \star (\otimes_{p\, P_i \in P} P_i) \otimes_{pu} (\otimes_{u U_i \in U} U_i)$ sous la forme $F \star P \otimes_{pu} U$, c'est-à-dire considérons la combinaison de fonctions locales du même type comme implicite. Le problème de décomposabilité à résoudre consiste à calculer des quantités de la forme $\oplus_{u\, x} (P \otimes_{pu} U)$ en ne considérant que les fonctions locales de P^{+x} et U^{+x} (nous verrons par ailleurs que les faisabilités et les éliminations avec $\min$ ou $\max$ ne posent pas de problème du point de vue de la décomposabilité).

2) Un deuxième problème est que la réponse à une requête PFU telle que définie par l'équation 1 utilise plusieurs opérateurs de combinaison ($\wedge$, $\star$, $\otimes_p$, $\otimes_{pu}$, $\otimes_u$), alors qu'un algorithme générique d'élimination de variables est conçu pour fonctionner avec un seul opérateur de combinaison $\otimes$.

3) De manière similaire, l'algorithme **VE** utilise un seul opérateur d'élimination $\oplus$, alors que le cadre PFU en utilise plusieurs ($\min$, $\max$, $\oplus_u$).

Ces différents problèmes sont successivement résolus ci-après afin d'obtenir un algorithme générique d'élimination de variables multi-opérateur applicable au cadre PFU.

3.3. *Résolution du problème de décomposabilité*

Intéressons nous tout d'abord au problème de décomposabilité lié au calcul de formes du type $\oplus_{u\, x} (P \otimes_{pu} U)$. Pour résoudre ce problème, nous avons retenu deux ensembles de propriétés supplémentaires, parce qu'ils sont l'un ou l'autre satisfaits par la plupart des cadres classiques (probabilistes ou non) couverts par le cadre PFU. Ces deux ensembles, notés Ax^{SA} et Ax^{SG}, s'écrivent :

Ax^{SA} : "$\otimes_u$ distributif par rapport à $\oplus_u$ et $p \otimes_{pu} (u_1 \otimes_u u_2) = (p \otimes_{pu} u_1) \otimes_u u_2$"

Ax^{SG} : "$\otimes_u = \oplus_u$ sur E_u" (mais pas sur $E_u \cup \{\Diamond\}$)

Le premier axiome est noté Ax^{SA} comme "axiome du cas semi-anneau" car il confère à $(E_u, \oplus_u, \otimes_u)$ une structure de semi-anneau. Le second axiome est noté

3. Une telle quantité peut être rencontrée si l'on souhaite calculer l'espérance probabiliste de degrés de satisfaction combinés par un $\min$.

Ax^{SG} comme "axiome du cas semi-groupe" car il rend la structure $(E_u, \oplus_u, \otimes_u)$ en quelque sorte équivalente à la structure $(E_u, \oplus_u)$, qui est un semi-groupe. Le tableau 2 montre que ces deux axiomes couvrent à eux deux toutes les structures d'utilité espérée du tableau 1.

La proposition 1 montre que dès que l'un des deux axiomes est satisfait, le problème de décomposabilité est résolu. Autrement dit, Ax^{SA} et Ax^{SG} sont des conditions suffisantes de décomposabilité.

Proposition 1. *Soit $(S_p, S_u, \oplus_u, \otimes_{pu})$ une structure d'utilité espérée. Soit P et U des ensembles de fonctions locales de plausibilité et d'utilité respectivement. Alors, $\oplus_{ux}(P \otimes_{pu} U) = P^{-x} \otimes_{pu} (\oplus_{ux}(P^{+x} \otimes_{pu} U))$. De plus,*

– si l'axiome Ax^{SA} est verifié, alors

$$\bigoplus_{x}^{u} (P^{+x} \otimes_{pu} U) \quad = \quad U^{-x} \otimes_u (\bigoplus_{x}^{u} (P^{+x} \otimes_{pu} U^{+x})) \qquad [2]$$

– si l'axiome Ax^{SG} est vérifié, alors

$$\bigoplus_{x}^{u} (P^{+x} \otimes_{pu} U) \quad = \quad ((\bigoplus_{x}^{p} P^{+x}) \otimes_{pu} U^{-x}) \otimes_u (\bigoplus_{x}^{u} (P^{+x} \otimes_{pu} U^{+x})) \qquad [3]$$

Par conséquent, si Ax^{SA} ou Ax^{SG} est vérifié, il est possible de ne considérer que les fonctions de P^{+x} et U^{+x} pour éliminer x dans l'expression $\oplus_{ux}(P \otimes_{pu} U)$. Cette proposition peut être illustrée par les structures utilisées pour la satisfaction espérée probabiliste (ayant pour but l'optimisation de la probabilité que des contraintes soient satisfaites) et pour l'utilité additive espérée probabiliste. Dans le premier cas, qui satisfait Ax^{SA}, on peut écrire $\sum_x (P^{+x} \times U) = U^{-x} \times (\sum_x (P^{+x} \times U^{+x}))$. Dans le second cas, on peut écrire $\sum_x (P^{+x} \times U) = ((\sum_x P^{+x}) \times U^{-x}) + (\sum_x (P^{+x} \times U^{+x}))$.

Notons que les axiomes Ax^{SA} et Ax^{SG} recouvrent chacun à la fois des cadres probabilistes et non probabilistes. Ceci reflète le fait que l'important d'un point de vue algorithmique n'est pas la nature probabiliste ou non d'un cadre donné mais les propriétés algébriques des opérateurs de combinaison et d'élimination utilisés.

	E_p	E_u	$\otimes_u$	$\oplus_u$	$\otimes_{pu}$	Ax^{SA}	Ax^{SG}
1	$\mathbb{R}^+$	$\mathbb{R} \cup \{-\infty\}$	$+$	$+$	$\times$		$\surd$
2	$\mathbb{R}^+$	$\mathbb{R}^+$	$\times$	$+$	$\times$	$\surd$	
3	$[0,1]$	$[0,1]$	min	max	min	$\surd$	
4	$[0,1]$	$[0,1]$	min	min	$\max(1-p, u)$		$\surd$
5	$\mathbb{N} \cup \{\infty\}$	$\mathbb{N} \cup \{\infty\}$	$+$	min	$+$	$\surd$	
6	$\{t, f\}$	$\{t, f\}$	$\wedge$	$\vee$	$\wedge$	$\surd$	
7	$\{t, f\}$	$\{t, f\}$	$\wedge$	$\wedge$	$\rightarrow$		$\surd$
8	$\{t, f\}$	$\{t, f\}$	$\vee$	$\vee$	$\wedge$		$\surd$
9	$\{t, f\}$	$\{t, f\}$	$\vee$	$\wedge$	$\rightarrow$	$\surd$	

Tableau 2. *Structures d'utilité espérées saisfaisant Ax^{SA} ou Ax^{SG} ; les indices indiqués dans la 1ère colonne correspondent aux indices des lignes du tableau 1*

Le problème de décomposabilité étant résolu dans les deux grandes classes axiomatiques que sont Ax^{SA} et Ax^{SG}, les deux problèmes restants - existence de plusieurs opérateurs de combinaison et existence de plusieurs opérateurs d'élimination - sont traités d'abord dans le cas semi-anneau (Ax^{SA}) puis dans le cas semi-groupe (Ax^{SG}).

3.4. *Le cas semi-anneau*

Le cas semi-anneau (axiome Ax^{SA}) couvre aussi bien des cadres probabilistes tels que la satisfaction espérée probabiliste que des cadres non probabilistes tels que l'utilité espérée possibiliste optimiste.

3.4.1. *Vers un seul opérateur de combinaison...*

Lorsque l'axiome Ax^{SA} est satisfait, il est en fait possible de se ramener à une structure algébrique beaucoup plus simple en transformant l'espace des degrés de plausibilité E_p en l'espace des degrés d'utilité E_u via un morphisme, pour n'avoir à travailler que sur un seul espace E égal à E_u faisant intervenir un seul opérateur de combinaison $\otimes$ égal à $\otimes_u$. Plus précisément, l'axiome Ax^{SA} est d'une certaine manière équivalent à l'axiome suivant :

$$Ax^{SA'} : \begin{cases} (E_p, \preceq_p) = (E_u, \preceq_u) = (E, \preceq) \\ \otimes_p = \otimes_{pu} = \otimes_u = \otimes \\ \oplus_p = \oplus_u = \oplus \end{cases}$$

Ce résultat est défini formellement par la proposition 2.

Proposition 2. *Soit $S = ((E_p, \oplus_p, \otimes_p), (E_u, \otimes_u), \oplus_u, \otimes_{pu})$ une structure d'utilité espérée totalement ordonnée par $\preceq_u$. Soit $\phi : E_p \rightarrow E_u$ la fonction définie par $\phi(p) = p \otimes_{pu} 1_u$.*

(a) Si S satisfait $Ax^{SA'}$, alors S satisfait Ax^{SA}.

(b) Si S satisfait Ax^{SA}, soit $(E, \preceq) = (E_u, \preceq_u)$, $\oplus = \oplus_u$ et $\otimes = \otimes_u$. Alors,

- La structure $S' = ((E, \oplus, \otimes), (E, \otimes), \oplus, \otimes)$ est une structure d'utilité espérée totalement ordonnée qui satisfait $Ax^{SA'}$.

- Tout réseau PFU $\mathcal{N} = (V, G, P, F, U)$ sur S peut être transformé en un réseau PFU $(V, G, \{\phi(P_i) \mid P_i \in P\}, F, U)$ sur S'. Ce nouveau réseau est noté $\phi(\mathcal{N})$.

- Pour toute requête $Q = (Sov, \mathcal{N})$ sur un réseau PFU $\mathcal{N}$ défini sur S, le couple $Q' = (Sov, \phi(\mathcal{N}))$ définit une requête sur le réseau PFU $\phi(\mathcal{N})$. De plus, Q et Q' sont équivalentes dans le sens où $Ans(Q) = Ans(Q')$ et les politiques optimales pour les décisions sont les mêmes pour Q et Q'.

La proposition 2(a) montre que Ax^{SA} est un axiome plus faible que $Ax^{SA'}$. Réciproquement, la proposition 2(b) montre que si une structure d'utilité espérée satisfait Ax^{SA}, alors il est possible de retrouver $Ax^{SA'}$ grâce au morphisme $\phi : p \rightarrow$

$p \otimes_{pu} 1_u$.[4] Ce morphisme transforme une requête $Q = (Sov, \mathcal{N})$ sur un réseau $\mathcal{N}$ en une requête équivalente sur le réseau $\phi(\mathcal{N})$.

Lorsque l'axiome $Ax^{SA'}$ est satisfait, la réponse à une requête prend la forme $Ans(Q) = Sov((\wedge_{F_i \in F} F_i) \star (\otimes_{\varphi \in P \cup U} \varphi))$, où $\otimes = \otimes_u$ possède un élément neutre $1_E = 1_u$. En outre, au lieu d'exprimer les faisabilités sur $\{t, f\}$, nous pouvons les exprimer sur $\{1_E, \Diamond\}$ en associant la valeur 1_E à t et la valeur $\Diamond$ à f. Cet artifice technique préserve la valeur d'une requête car $t \star u = 1_E \otimes u$ et $f \star u = \Diamond \otimes u$. La réponse à une requête Q devient

$$Ans(Q) = Sov(\bigotimes_{\varphi \in P \cup F \cup U} \varphi) \qquad [4]$$

Le passage de Ax^{SA} à $Ax^{SA'}$ permet donc de n'utiliser qu'un seul opérateur de combinaison $\otimes$. Le seul problème restant est alors l'existence de plusieurs opérateurs d'élimination.

3.4.2. *Gestion des différents opérateurs d'élimination*

La séquence opérateur-variable(s) Sov de l'équation 4 fait intervenir les opérateurs d'élimination $\min$, $\max$ et/ou $\oplus$. Ces opérateurs satisfont certaines bonnes propriétés :

Proposition 3. *Soit* $((E_p, \oplus_p, \otimes_p), (E_u, \otimes_u), \oplus_u, \otimes_{pu})$ *une structure d'utilité espérée totalement ordonnée qui satisfait* $Ax^{SA'}$. *Soit* $(E, \oplus, \otimes) = (E_u, \oplus_u, \otimes_u)$. *Alors,* $(E \cup \{\Diamond\}, \min, \otimes)$, $(E \cup \{\Diamond\}, \max, \otimes)$ *et* $(E \cup \{\Diamond\}, \oplus, \otimes)$ *sont tous trois des semi-anneaux commutatifs.*

La proposition précédente assure que pour tout ensemble de fonctions locales Φ et pour tout opérateur d'élimination $op \in \{\min, \max, \oplus\}$ utilisé, une élimination $op_x \Phi$ est toujours décomposable sous la forme $\Phi^{-x} \otimes (op_x \Phi^{+x})$.

Afin de résoudre le problème lié à l'existence de plusieurs opérateurs d'élimination qui *en général* ne peuvent pas être commutés sans changer la réponse à la requête, il suffit d'imposer des contraintes sur l'ordre d'élimination des variables : si deux variables x et y appartiennent à deux paires (opérateur d'élimination-ensemble de variables) différentes, op_S et $op'_{S'}$ respectivement, et si op_S est située à gauche de $op'_{S'}$ dans Sov, alors x doit être éliminée après y ; si elles appartiennent à une même paire, leur ordre d'élimination est indifférent. Les éliminations adjacentes utilisant un même opérateur d'élimination peuvent bien sûr être fusionnées avant de définir de telles contraintes sur l'ordre d'élimination.

Les problèmes de décomposabilité, d'existence de plusieurs opérateurs de combinaison et d'existence de plusieurs opérateurs d'élimination sont par conséquent résolus dans le cas semi-anneau. Ceci nous conduit à l'algorithme 2, un algorithme

4. Notons qu'il existe des structures qui satisfont Ax^{SA} mais qui ne satisfont pas directement $Ax^{SA'}$. C'est le cas de la structure donnée à la ligne 9 du tableau 2, qui satisfait $Ax^{SA'}$ uniquement après application du morphisme ϕ.

d'élimination de variables multi-opérateur noté **MVE** pour "**M**ulti-operator **V**ariable **E**limination". Le premier appel est **MVE**$(Sov, \otimes, P \cup F \cup U)$.

Algorithme 2 : Un algorithme d'élimination de variables multi-opérateur générique (Sov : séquence d'éliminations, $\circledast$: opérateur de combinaison, Φ : ensemble de fonctions locales portant sur les variables qui apparaissent dans Sov)

MVE$(Sov, \circledast, \Phi)$
début
  tant que $Sov \neq \emptyset$ **faire**
    $Sov'.(op, S) \leftarrow Sov$
    choisir $x \in S$
    si $S = \{x\}$ **alors** $Sov \leftarrow Sov'$ **sinon** $Sov \leftarrow Sov'.(op, S - \{x\})$
    $\varphi_0 \leftarrow op_x \left(\circledast_{\varphi \in \Phi^{+x}} \varphi\right)$
    $\Phi \leftarrow \Phi^{-x} \cup \{\varphi_0\}$
  retourner $\circledast_{\varphi \in \Phi} \varphi$
fin

Proposition 4. *Etant donnée une requête* $Q = (Sov, (V, G, P, F, U))$ *dans le cas semi-anneau,* ***MVE***$(Sov, \otimes, P \cup F \cup U)$ *renvoie* $Ans(Q)$*. De plus, les politiques optimales pour les décisions définies par* ***MVE***$(Sov, \otimes, P \cup F \cup U)$ *sont les mêmes que les politiques optimales définies par* $Ans(Q)$*.*

Exemple. Illustrons l'utilisation de l'algorithme **MVE** dans le cas semi-anneau à partir du problème des capteurs pour la séquence $Sov = \max_{te} \oplus_{urte} \max_{rp} \oplus_{uecs,ec_A,ec_B}$. Le tableau suivant indique les fonctions locales créées au cours des éliminations ainsi que l'ensemble des fonctions locales à chaque étape. Les variables sont ici éliminées dans l'ordre $ec_B, ec_A, ecs, rp, rte, te$ qui respecte bien les contraintes sur l'ordre d'élimination imposées par Sov.

Elimination	Ensemble des fonctions restantes après élimination
$\oplus_{ec_B}$	$\Phi = \{P_\emptyset, P_{rte,te}, P_{rte,ec_A,te}, F_{rp,te}, U_{te}, U_{rp}, U_{rp,ecs}, \varphi_1\}$ avec $\varphi_1 = \oplus_{ec_B}(P_{rte,ec_B,te} \otimes P_{ec_A,ec_B} \otimes P_{ecs,ec_A,ec_B,rp})$
$\oplus_{ec_A}$	$\Phi = \{P_\emptyset, P_{rte,te}, F_{rp,te}, U_{te}, U_{rp}, U_{rp,ecs}, \varphi_2\}$ avec $\varphi_2 = \oplus_{ec_A}(P_{rte,ec_A,te} \otimes \varphi_1)$
$\oplus_{ecs}$	$\Phi = \{P_\emptyset, P_{rte,te}, F_{rp,te}, U_{te}, U_{rp}, \varphi_3\}$ avec $\varphi_3 = \oplus_{ecs}(U_{rp,ecs} \otimes \varphi_2)$
$\max_{rp}$	$\Phi = \{P_\emptyset, P_{rte,te}, U_{te}, \varphi_4\}$ avec $\varphi_4 = \max_{rp}(F_{rp,te} \otimes U_{rp} \otimes \varphi_3)$
$\oplus_{rte}$	$\Phi = \{P_\emptyset, U_{te}, \varphi_5\}$ avec $\varphi_5 = \oplus_{rte}(P_{rte,te} \otimes \varphi_4)$
$\max_{te}$	$\Phi = \{P_\emptyset, \varphi_6\}$ avec $\varphi_6 = \max_{te}(U_{te} \otimes \varphi_5)$

D'un point de vue plus global, l'algorithme MVE utilise simplement les propriétés algébriques des opérateurs pour décomposer automatiquement les calculs sous la forme :

$$Ans(Q) = P_{\emptyset} \otimes \max_{te} \left(U_{te} \otimes \left(\bigoplus_{rte} \left(P_{rte,te} \otimes \max_{rp} \left(F_{rp,te} \otimes U_{rp} \otimes \left(\bigoplus_{ecs} (U_{rp,ecs} \otimes \right. \right. \right. \right. \right.$$
$$\left. \left. \left. \left. \left. \left(\bigoplus_{ec_A} \left(P_{rte,ec_A,te} \otimes \left(\bigoplus_{ec_B} (P_{rte,ec_B,te} \otimes P_{ec_A,ec_B} \otimes P_{ecs,ec_A,ec_B,rp}) \right) \right) \cdots \right) \right. \right. \right. \right. \right.$$

Une telle décomposition est valide pour toutes les structures d'utilité espérée associées au cas semi-anneau : elle est valide, suivant la structure algébrique utilisée et le contenu des fonctions locales, pour maximiser la probabilité que les capteurs fonctionnent après réparation éventuelle (avec $\oplus = +$ et $\otimes = \times$), pour optimiser une utilité espérée possibiliste optimiste (avec $\oplus = \max$ et $\otimes = \min$), ou encore pour savoir s'il existe un monde dans lequel un ensemble de contraintes est satisfait (avec $\oplus = \vee$, $\otimes = \wedge$ et $f \prec t$).

3.5. *Le cas semi-groupe*

Le second cas à traiter est le cas semi-groupe (axiome Ax^{SG} : "$\otimes_u = \oplus_u$ sur E_u"). Ce cas couvre aussi bien des cadres probabilistes tels que l'utilité espérée probabiliste additive que des cadres non probabilistes tels que l'utilité espérée possibiliste pessimiste. Comme dans le cas semi-anneau, nous avons à régler les deux problèmes que sont l'existence de plusieurs opérateurs de combinaison et l'existence de plusieurs opérateurs d'élimination.

3.5.1. *Vers un seul opérateur de combinaison...*

Le cas semi-groupe nécessite un peu plus de travail que le cas semi-anneau, principalement car l'équation 3 ne crée pas uniquement une nouvelle fonction d'utilité résultant de l'élimination de x : elle crée premièrement une nouvelle fonction de plausibilité $\oplus_{p_x} P^{+x}$ qui doit être combinée avec toutes les fonctions d'utilité de U^{-x}, et deuxièmement une nouvelle fonction d'utilité $\oplus_{u_x} (P^{+x} \otimes_{pu} U^{+x})$.

Afin d'exploiter une forme générale qui ne varie pas durant les éliminations et qui utilise un seul opérateur de combinaison, comme le fait un algorithme d'élimination de variables classique, une solution consiste à travailler sur des couples "(fonction de plausibilité , fonction d'utilité)" appelés des *potentiels* (Ndilikilikesha, 1994).

Définition 7. *Un* potentiel *est une paire (P_0, U_0) composée d'une fonction de plausibilité P_0 et d'une fonction d'utilité U_0. Deux opérateurs sont introduits sur les paires*

plausibilité-utilité : un opérateur de combinaison $\boxtimes$ *et un opérateur d'élimination* $\boxplus$ *définis par*[5] [6]

$$(p_1, u_1) \boxtimes (p_2, u_2) = (p_1 \otimes_p p_2, (p_1 \otimes_{pu} u_2) \otimes_u (p_2 \otimes_{pu} u_1))$$
$$(p_1, u_1) \boxplus (p_2, u_2) = (p_1 \oplus_p p_2, u_1 \oplus_u u_2)$$

Enfin, un ordre partiel $\preceq$ *est défini sur les paires plausibilité-utilité par*

$$((p, u_1) \preceq (p, u_2)) \leftrightarrow (u_1 \preceq_u u_2)$$

Dans ce qui suit, nous considérons également chaque fonction de faisabilité comme un potentiel, en associant la valeur $(1_p, 1_u)$ (qui est un élément neutre pour $\boxtimes$) au degré de faisabilité t et la valeur $\Diamond$ au degré de faisabilité f.

La proposition suivante montre comment les potentiels permettent de calculer la réponse à une requête en utilisant un seul opérateur de combinaison, $\boxtimes$. L'idée consiste à associer le potentiel $(P_i, 1_u)$ à chaque fonction de plausibilité P_i et à associer le potentiel $(1_p, U_i)$ à chaque fonction d'utilité U_i.

Proposition 5. *Soit* $Q = (Sov, (V, G, P, F, U))$ *une requête sur un réseau PFU défini sur une structure d'utilité espérée totalement ordonnée et qui satisfait* Ax^{SG}. *Soit* $T(Sov)$ *la séquence de paires opérateur-variable(s) obtenue à partir de* Sov *en remplaçant les éliminations avec* $\oplus_u$ *sur des variables d'environnement par des éliminations avec* $\boxplus$. *Soit* Π_0 *l'ensemble de potentiels*

$$\Pi_0 = \{(P_i, 1_u), P_i \in P\} \cup F \cup \{(1_p, U_i), U_i \in U\}$$

Alors, $T(Sov)\,(\boxtimes_{\varphi \in \Pi_0} \varphi) = \begin{cases} (1_p, Ans(Q)) \text{ si } Ans(Q) \neq \Diamond \\ \Diamond \text{ sinon} \end{cases}$

Le problème de l'existence de plusieurs opérateurs de combinaison est ainsi résolu grâce aux potentiels.

3.5.2. *Gestion des différents opérateurs d'élimination*

La proposition 6 montre qu'en plus de permettre l'utilisation d'un seul opérateur de combinaison, les potentiels bénéficient de certaines bonnes propriétés concernant les différents opérateurs d'élimination utilisés.

---

5. La définition de $(p_1, u_1) \boxtimes (p_2, u_2)$ peut sembler un peu surprenante au premier abord. En fait, p_1 correspond informellement à un degré de plausibilité déjà "intégré" dans u_1 mais pas dans les autres utilités, d'où la combinaison $p_1 \otimes_{pu} u_2$. De même, p_2 est déjà "intégré" dans u_2 et doit pondérer toutes les autres utilités, d'où la combinaison $p_2 \otimes_{pu} u_1$.

6. Les opérateurs $\boxplus$ et $\boxtimes$ définis ici diffèrent de ceux introduits dans (Ndilikilikesha, 1994) qui s'appliquent par ailleurs uniquement au cadre probabiliste des diagrammes d'influence. L'opérateur $\boxplus$ défini dans (Ndilikilikesha, 1994) par $(p_1, u_1) \boxplus (p_2, u_2) = (p_1 + p_2, \frac{p_1 \cdot u_1 + p_2 \cdot u_2}{p_1 + p_2})$ utilise notamment une opération de division, alors que notre définition n'en fait pas usage.

Proposition 6. *Considérons une structure d'utilité espérée totalement ordonnée qui satisfait Ax^{SG}. Alors,*

– $\boxplus$ et $\boxtimes$ sont commutatifs, associatifs, $(1_p, 1_u)$ est un élément neutre pour $\boxtimes$ et $\Diamond$ est un élément neutre pour $\boxplus$ et absorbant pour $\boxtimes$;

– $\boxtimes$ est distributif par rapport à $\boxplus$. Ceci implique que pour tout ensemble de potentiels Π, $\boxplus_x(\Pi) = \Pi^{-x} \boxtimes (\boxplus_x(\Pi^{+x}))$.

– $\boxtimes$ satisfait une propriété de distributivité restreinte par rapport à $\min$ et $\max$: pour tout ensemble de potentiels Π tel que $x \notin sc(P_0)$ pour tout $(P_0, U_0) \in \Pi$,

- $\max_x(\Pi)$ *existe et* $\max_x(\Pi) = \Pi^{-x} \boxtimes \max_x(\Pi^{+x})$,
- $\min_x(\Pi)$ *existe et* $\min_x(\Pi) = \Pi^{-x} \boxtimes \min_x(\Pi^{+x})$.

La proposition précédente ne dit cependant pas que $\boxtimes$ est distributif par rapport à tous les opérateurs d'élimination utilisables. L'opérateur $\boxtimes$ n'est en effet pas distributif par rapport à $\min$ et $\max$, essentiellement car seul un ordre partiel est défini sur les paires plausibilité-utilité (par exemple $\max((0.2, 4), (0.6, 3))$ n'existe pas). Heureusement, nous allons voir que la propriété de distributivité restreinte indiquée dans la proposition 6 est suffisante pour utiliser un algorithme d'élimination de variables dès lors que l'on modifie légèrement la définition de Φ^{+x} pour un ensemble de fonctions Φ de la manière suivante : [7]

Si x est la dernière variable d'une composante c à être éliminée, alors $\Phi^{+x} = \{\varphi \in \Phi \mid x \in sc(\varphi)\} \cup (\Phi \cap Fact(c))$.

Informellement, cette nouvelle définition de Φ^{+x} signifie que lorsque x est la dernière variable de sa composante c à être éliminée, on ajoute à Φ^{+x} les fonctions de $Fact(c)$ qui n'auraient pas encore été considérées au cours des éliminations. Ces fonctions correspondent en fait exactement aux fonctions de $Fact(c)$ dont la portée est incluse dans $pa(c)$. Cette modification est nécessaire pour pouvoir utiliser certaines conditions de normalisation et assurer la bonne définition d'opérations de minimisation ou de maximisation sur les potentiels, qui sont comparables uniquement si leur partie plausibilité est constante. La définition $\Phi^{-x} = \Phi - \Phi^{+x}$ est conservée.

Cette légère modification de la définition de Φ^{+x} permet d'aboutir à la proposition 7, qui montre la possibilité d'utiliser l'algorithme **MVE** pour calculer la réponse à une requête dans le cas semi-groupe.

Proposition 7. *Soit $Q = (Sov, (V, G, P, F, U))$ une requête dans le cas semi-groupe. Soit $T(Sov)$ la séquence de paires opérateur-variable(s) obtenue à partir de Sov en remplaçant les éliminations avec $\oplus_u$ sur des variables d'environnement par des éliminations avec $\boxplus$. Soit Π_0 l'ensemble de potentiels*

$$\Pi_0 = \{(P_i, 1_u), P_i \in P\} \cup F \cup \{(1_p, U_i), U_i \in U\}$$

7. Nous considérons que si $P_i \in Fact(c)$ pour une composante c alors le potentiel $(P_i, 1_u)$ appartient à $Fact(c)$.

Alors, $\mathbf{MVE}(T(Sov), \boxtimes, \Pi_0)$ *renvoie* $(1_p, Ans(Q))$ *si* $Ans(Q) \neq \Diamond$ *et* $\Diamond$ *sinon. De plus, les politiques optimales pour les décisions définies par* $\mathbf{MVE}(T(Sov), \boxtimes, \Pi_0)$ *sont les mêmes que les politiques optimales définies par* $Ans(Q)$.

L'algorithme MVE est par conséquent un algorithme générique utilisable aussi bien dans le cas semi-anneau que dans le cas semi-groupe.

Exemple. Reprenons l'exemple du problème des capteurs, mais cette fois avec une structure d'utilité espérée du type semi-groupe. L'ensemble des potentiels avant la première élimination est :

$$\begin{aligned}\Pi \;=\; & \{(P_\emptyset, 1_u), (P_{ec_A,ec_B}, 1_u), (P_{rte,te}, 1_u), (P_{rte,ec_A,te}, 1_u), (P_{rte,ec_B,te}, 1_u),\\ & (P_{ecs,ec_A,ec_B,rp}, 1_u), F_{rp,te}, (1_p, U_{te}), (1_p, U_{rp}), (1_p, U_{rp,ecs})\}\end{aligned}$$

Les potentiels créés à chaque étape sont fournis dans le tableau suivant :

Elim.	Potentiel créé après l'élimination de variable
$\boxplus_{ec_B}$	$\pi_1 = \boxplus_{ec_B}((P_{rte,ec_B,te}, 1_u) \boxtimes (P_{ec_A,ec_B}, 1_u) \boxtimes (P_{ecs,ec_A,ec_B,rp}, 1_u))$ $= (P_1, 1_u)$ avec $P_1 = \oplus_{p_{ec_B}}(P_{rte,ec_B,te} \otimes_p P_{ec_A,ec_B} \otimes_p P_{ecs,ec_A,ec_B,rp})$
$\boxplus_{ec_A}$	$\pi_2 = \boxplus_{ec_A}((P_\emptyset, 1_u) \boxtimes (P_{rte,ec_A,te}, 1_u) \boxtimes \pi_1) = (P_2, 1_u)$ avec $P_2 = \oplus_{p_{ec_A}}(P_\emptyset \otimes_p P_{rte,ec_A,te} \otimes_p P_1)$
$\boxplus_{ecs}$	$\pi_3 = \boxplus_{ecs}((1_p, U_{rp,ecs}) \boxtimes \pi_2) = (P_3, U_3)$ avec $P_3 = \oplus_{p_{ecs}} P_2$ et $U_3 = \oplus_{uecs}(P_2 \otimes_{pu} U_{rp,ecs})$
$\max_{rp}$	$\pi_4 = \max_{rp}(F_{rp,te} \boxtimes (1_p, U_{rp}) \boxtimes \pi_3) = (P_4, U_4)$ avec $P_4 = P_3$ et $U_4 = \max_{rp}(F_{rp,te} \star (P_3 \otimes_{pu} U_{rp}) \otimes_u (1_p \otimes_{pu} U_3))$
$\boxplus_{rte}$	$\pi_5 = \boxplus_{rte}((P_{rte,te}, 1_u) \boxtimes \pi_4) = (P_5, U_5)$ avec $P_5 = \oplus_{p_{rte}}(P_{rte,te} \otimes_p P_4)$ et $U_5 = \oplus_{urte}(P_{rte,te} \otimes_{pu} U_4)$
$\max_{te}$	$\pi_6 = \max_{te}((1_p, U_{te}) \boxtimes \pi_5) = (P_6, U_6) = (1_p, Ans(Q))$ avec $U_6 = \max_{te}(U_{te} \otimes_u U_5)$

Quelques commentaires s'imposent concernant les différentes étapes d'éliminations. L'élimination de ec_B se fait naturellement en éliminant ec_B sur la combinaison de tous les potentiels qui font intervenir ec_B. Pour l'élimination de ec_A, on considère les potentiels qui portent sur ec_A et le potentiel $(P_\emptyset, 1_u)$. Ce dernier potentiel est considéré à cause de la modification de la définition de Φ^{+x} : en effet, $P_\emptyset$ est un facteur de la composante $\{ec_A, ec_B\}$ et ec_A est la dernière variable de cette composante à être éliminée. Nous verrons plus tard l'intérêt d'avoir considéré $(P_\emptyset, 1_u)$ à ce niveau là.

La variable ecs est ensuite éliminée, puis on effectue une opération de maximisation sur rp. Pour que cette opération de maximisation soit bien définie, il faut que la partie plausibilité du potentiel $F_{rp,te} \boxtimes (1_p, U_{rp}) \boxtimes \pi_3$ ne dépendent pas de la valeur de rp. Cette condition est bien satisfaite car grâce aux conditions de normalisation, la partie plausibilité P_3 de π_3 vaut

$\oplus_{p_{ecs}} P_2$
$= \oplus_{p_{ecs,ec_A,ec_B}}(P_\emptyset \otimes_p P_{rte,ec_A,te} \otimes_p P_{rte,ec_B,te} \otimes_p P_{ec_A,ec_B} \otimes_p P_{ecs,ec_A,ec_B,rp})$
$= \oplus_{p_{ec_A,ec_B}}(P_\emptyset \otimes_p P_{rte,ec_A,te} \otimes_p P_{rte,ec_B,te} \otimes_p P_{ec_A,ec_B})$

Le deuxième point est que l'on s'autorise à écrire $\max_{rp}(F_{rp,te} \boxtimes (1_p, U_{rp}) \boxtimes \pi_3)$ sous la forme $(P_4, \max_{rp}(F_{rp,te} \star (\ldots)))$, ou autrement dit on s'autorise à "rentrer" la fonction de faisabilité $F_{rp,te}$ dans la partie utilité du potentiel créé. Ceci est possible car, du fait des conditions de normalisation, il existe une valeur faisable pour rp quelle que soit la valeur de te.

La variable rte est ensuite éliminée et enfin on effectue une maximisation sur te. On est comme auparavant assuré que grâce aux conditions de normalisation, et notamment grâce au fait que $(P_\emptyset, 1_u)$ a été intégré au départ à la partie plausibilité du potentiel courant, la partie plausibilité du potentiel à optimiser est constante et égale à 1_p.

D'un point de vue plus global, tout se passe en fait comme si l'utilisation combinée des potentiels et de l'algorithme **MVE** décomposait le calcul de $Ans(Q)$ sous la forme :

$$\max_{te}\left((P_5 \otimes_{pu} U_{te}) \otimes_u \left(\bigoplus_{rte}{}_u\, P_{rte,te} \otimes_{pu} \max_{rp}\left(\begin{array}{c} F_{rp,te} \star (P_3 \otimes_{pu} U_{rp}) \\ \otimes_u(\bigoplus_{ecs}{}_u(P_2 \otimes_{pu} U_{rp,ecs})) \end{array}\right)\right)\right)$$

$$\text{avec } \left\{\begin{array}{l} P_5 = \oplus_{p_{rte}}(P_{rte,te} \otimes_p P_3) \\ P_3 = \oplus_{p_{ecs}} P_2 \\ P_2 = \oplus_{p_{ec_A}}(P_\emptyset \otimes_p P_{rte,ec_A,te} \otimes_p P_1) \\ P_1 = \oplus_{p_{ec_B}}(P_{rte,ec_B,te} \otimes_p P_{ec_A,ec_B} \otimes_p P_{ecs,ec_A,ec_B,rp}) \end{array}\right.$$

Une telle décomposition est valide aussi bien si l'on souhaite optimiser une utilité espérée probabiliste additive (avec $\oplus_p = \oplus_u = \otimes_u = +$ et $\otimes_p = \otimes_{pu} = \times$) que si l'on souhaite optimiser une utilité espérée possibiliste pessimiste (avec $\oplus_p = \max$, $\oplus_u = \otimes_p = \otimes_u = \min$ et $\otimes_{pu} = \max(1-p, u)$), ou encore si l'on souhaite savoir si un ensemble de contraintes est satisfait dans tous les mondes possibles (avec $\oplus_p = \vee$, $\oplus_u = \otimes_p = \otimes_u = \wedge$, $\otimes_{pu} = \rightarrow$ et $f \prec t$).

3.6. *Cas général*

Il existe des structures d'utilité espérée qui ne satisfont ni Ax^{SA} ni Ax^{SG}. C'est le cas pour des structures qui utilisent $\oplus_u = +$, $\otimes_{pu} = \times$ et $\otimes_u = \min$, et qui induisent des calculs de la forme $\sum_x(P_{xy} \cdot \min(U_x, U_{zt}))$. Lorsque ni Ax^{SA} ni Ax^{SG} n'est satisfait, on peut, en combinant toutes les fonctions d'utilité pour obtenir une unique fonction d'utilité globale $U_0 = \otimes_{u U_i \in U} U_i$ et en utilisant un morphisme similaire à celui utilisé dans le cas semi-anneau, se ramener à un calcul de la forme :

$$Ans(Q) = Sov(\bigotimes_{\varphi \in P \cup F \cup \{U_0\}} \varphi) \qquad [5]$$

Le cas général est donc un cas particulier du cas semi-anneau *à condition* d'agréger toutes les fonctions locales d'utilité pour obtenir une unique fonction globale d'utilité.

L'algorithme **MVE** peut donc à nouveau être utilisé, avec **MVE**$(Sov, \otimes, P \cup F \cup \{U_0\})$ comme premier appel.

Le tableau 3 résume l'utilisation de l'algorithme **MVE** pour répondre à une requête PFU. Cet algorithme se révèle être un algorithme unifié utilisable aussi bien pour des cadres de décision dans l'incertain probabiliste que pour des cadres de décision dans l'incertain possibiliste. Cette unification est possible car ce sont les propriétés élémentaires des opérateurs de combinaison et d'élimination qui importent d'un point de vue algorithmique.

CAS	PREMIER APPEL
semi-anneau	**MVE**$(Sov, \otimes, P \cup F \cup U)$
semi-groupe	**MVE**$(T(Sov), \boxtimes, \{(P_i, 1_u), P_i \in P\} \cup F \cup \{(1_p, U_i), U_i \in U\})$
cas général	**MVE**$(Sov, \otimes, P \cup F \cup \{U_0\})$, avec $U_0 = \otimes_{uU_i \in U} U_i$

Tableau 3. *Utilisation de l'algorithme **MVE**, un algorithme générique unifié d'élimination de variables multi-opérateur*

3.7. *Comparaison avec l'existant*

Par rapport aux algorithmes classiques d'élimination de variables pour la décision dans l'incertain probabiliste (Ndilikilikesha, 1994), l'algorithme **MVE** utilise des potentiels *sans faire appel à des opérations de division*. Il permet ainsi d'économiser des calculs inutiles. Il est de plus capable de manipuler autre chose que des gains et des coûts additifs : par exemple, il peut être utilisé pour calculer des décisions maximisant la probabilité que des contraintes soient satisfaites. Cette flexibilité est due au fait que l'algorithme **MVE** s'applique aussi bien à des structures de type semi-anneau qu'à des structures de type semi-groupe. Par rapport aux algorithmes possibilistes d'élimination de variables utilisant les *Valuation Based Systems* (VBS (Shenoy, 1992) [8]), l'algorithme **MVE** apporte un gain double. Premièrement, il peut traiter des possibilités qualitatives ($(\oplus_p, \otimes_p) = (\max, \min)$) alors que l'approche VBS définie dans (Shenoy, 1992), qui impose une condition de normalisation sur les combinaisons de fonctions locales, ne le peut pas. Dans notre cas, cette normalisation après combinaison est inutile : ce qui compte est uniquement que le résultat obtenu après toutes les éliminations soit le bon, ce qui est garanti. Deuxièmement, l'algorithme **MVE** peut manipuler des utilités espérées possibilistes, et non uniquement des possibilités. Ceci est le résultat du caractère multi-opérateur de l'algorithme **MVE**, qui utilise plusieurs opérateurs pour éliminer les différents types de variables et pour combiner les différents types d'informations au sein des potentiels (lorsque les potentiels sont utilisés).

8. La notion de fonction locale est équivalente à la notion de *valuation* utilisée dans les VBS.

4. Evaluation de la complexité théorique

Cette section fournit des bornes supérieures sur la complexité temporelle et spatiale de l'algorithme unifié **MVE**. Ces bornes sont génériques car valables pour tous les formalismes couverts par le cadre PFU.

4.1. *Complexité d'un algorithme classique d'élimination de variables*

Une borne supérieure sur la complexité théorique de l'algorithme classique d'élimination de variables **VE** peut être définie en fonction d'un paramètre appelé *largeur induite* (Dechter *et al.*, 2001; Dechter, 2003). Ce paramètre est aussi connu sous le nom de largeur d'arbre (Robertson *et al.*, 1986) ou de *k-tree number* (Arnborg, 1985). Etant donnée une requête mono-opérateur de la forme $\oplus_V(\otimes_{\varphi\in\Phi})$, la largeur induite est définie à partir de l'hypergraphe $\mathcal{G} = (V, H)$ dont l'ensemble des hyper-arêtes $H = \{sc(\varphi) \,|\, \varphi \in \Phi\}$ représente les portées des fonctions locales de Φ. Cette largeur induite est introduite formellement ci-après.

Définition 8. *Soit $\mathcal{G} = (V, H)$ un hypergraphe. Soit o un ordre d'élimination sur V, c'est-à-dire une bijection de $\{1, \ldots, |V|\}$ dans V. o peut être utilisé pour générer une séquence d'hypergraphes $\mathcal{G}_1, \ldots, \mathcal{G}_{n+1}$ (avec $n = |V|$) telle que $\mathcal{G}_1 = \mathcal{G}$ et si $\mathcal{G}_k = (V_k, H_k)$ et si $x = o(k)$ est la k-ème variable éliminée dans o, alors $\mathcal{G}_{k+1} = (V_k - \{x\}, (H_k - H_k^{+x}) \cup \{h_{k+1}\})$, avec H_k^{+x} l'ensemble des hyper-arêtes de H_k impliquant la variable x et $h_{k+1} = (\cup_{h\in H_k^{+x}} h) - \{x\}$ l'hyper-arête créée de l'étape k à l'étape $k+1$ (étape d'élimination de variable). La largeur induite de $\mathcal{G}$ pour l'ordre d'élimination o, notée $w_{\mathcal{G}}(o)$ est égale à la taille maximale des hyper-arêtes créées, c'est-à-dire $w_{\mathcal{G}}(o) = \max_{k\in\{1,\ldots,n\}} |h_{k+1}|$.*

Informellement, l'hyper-arête h_{k+1} créée de l'étape k à l'étape $k+1$ est obtenue en considérant l'ensemble H_k^{+x} des hyper-arêtes de $\mathcal{G}_k$ qui "dépendent" de x et en "reliant" toutes les variables impliquées dans H_k^{+x} (sauf x). Ceci représente le fait que dans l'algorithme **VE**, l'élimination de x crée une nouvelle fonction locale de portée h_{k+1}. Un résultat connu est que si l'on suppose que chaque opérateur binaire utilisé renvoie un résultat en temps constant et que chaque lecture et écriture mémoire est en temps constant, [9] alors les complexités temporelle et spatiale de l'algorithme **VE** sont exponentielles en la largeur induite $w_{\mathcal{G}}(o)$, plus précisément en $O(|\Phi| \cdot d^{1+w_{\mathcal{G}}(o)})$ avec d la taille du plus grand domaine d'une variable.

La largeur induite d'un hypergraphe $\mathcal{G}$, notée $w_{\mathcal{G}}$, est égale à la largeur minimale induite par un ordre d'élimination sur V. Elle correspond au nombre mini-

9. En réalité, un opérateur retourne un résultat en un temps fonction de la taille de ses arguments et lire une valeur $\varphi(A)$ pour une fonction locale φ représentée par une table se fait en temps $O(|V| \cdot log(d))$. Nous adoptons ici la même convention que dans (Papadimitriou, 1994), qui ne modifie pas la classe de complexité de l'algorithme et peut être simplement relâchée en ajoutant le facteur $|V| \cdot log(d)$ aux résultats de complexités temporelles. Nous devrions donc dire que l'algorithme **VE** a une complexité temporelle $O(|V| \cdot log(d) \cdot |\Phi| \cdot d^{1+w_{\mathcal{G}}(o)})$.

mal de variables à considérer simultanément par un algorithme d'élimination de variables lorsqu'un ordre d'élimination optimal est utilisé. Le problème de décision associé à la recherche d'un ordre d'élimination optimal est cependant un problème NP-complet (Arnborg, 1985). Si l'ordre d'élimination utilisé par l'algorithme **VE** est optimal, alors il a une complexité $O(|\Phi| \cdot d^{1+w_{\mathcal{G}}})$.

4.2. *Largeur induite contrainte*

La largeur induite peut être utilisée pour quantifier la complexité d'un algorithme d'élimination de variables sans contraintes sur l'ordre d'élimination. Afin de définir la complexité de l'algorithme **MVE**, dans lequel l'ordre d'élimination des variables est contraint par la séquence Sov, nous utilisons un paramètre légèrement différent connu sous le nom de *largeur induite contrainte* (Jensen *et al.*, 1994; Park *et al.*, 2004).

Définition 9. *Soit $\mathcal{G} = (V, H)$ un hypergraphe et soit $\preceq$ un ordre partiel sur V. La largeur induite contrainte $w_{\mathcal{G}}(\preceq)$ de $\mathcal{G}$ avec contraintes sur l'ordre d'élimination données par $\preceq$ ("$x \prec y$" signifie "y doit être éliminée avant x") est définie par $w_{\mathcal{G}}(\preceq) = \min_{o \in \mathcal{O}} w_{\mathcal{G}}(o)$ où $\mathcal{O}$ est l'ensemble des ordres totaux $\preceq'$ sur V qui satisfont $(x \preceq y) \rightarrow (x \preceq' y)$.*

Les contraintes sur l'ordre d'élimination induites par la séquence d'éliminations de variables Sov peuvent être formellement définies de la manière suivante :

Définition 10. *Soit $Q = (Sov, \mathcal{N})$ une requête sur un réseau PFU telle que $Sov = (op_1, S_1) \cdot (op_2, S_2) \cdots (op_q, S_q)$. L'*ordre partiel *$\preceq_{Sov}$ induit par Sov est donné par $S_1 \prec_{Sov} S_2 \prec_{Sov} \ldots \prec_{Sov} S_q$. Cet ordre partiel force les variables de S_i à être éliminées après les variables de S_j dès que $i < j$.*

Par exemple, l'ordre partiel induit par la séquence d'élimination $Sov = \min_{x_1,x_2} \sum_{x_3,x_4} \max_{x_5}$ est défini par $\{x_1, x_2\} \prec_{Sov} \{x_3, x_4\} \prec_{Sov} x_5$.

Proposition 8. *Pour un ordre d'élimination o donné, l'algorithme **MVE**$(Sov, \circledast, \Phi)$ a une complexité spatiale et temporelle $O(|\Phi| \cdot d^{1+w_{\mathcal{G}}(o)})$, avec d la taille maximale du domaine des variables de V et $\mathcal{G}$ l'hypergraphe $(V, \{sc(\varphi) \,|\, \varphi \in \Phi\})$. Avec un ordre d'élimination optimal, la complexité spatiale et temporelle est $O(|\Phi| \cdot d^{1+w_{\mathcal{G}}(\preceq_{Sov})})$.*

Etant donnée une requête $Q = (Sov, \mathcal{N})$ sur un réseau PFU $\mathcal{N} = (V, G, P, F, U)$, la proposition 8 implique les résultats suivants : si un ordre d'élimination optimal est utilisé, alors la complexité théorique temporelle et spatiale de l'algorithme **MVE** est :

1) dans les cas semi-anneau et semi-groupe : $O(|P \cup F \cup U| \cdot d^{1+w_{\mathcal{G}}(\preceq_{Sov})})$, avec $\mathcal{G} = (V, \{sc(\varphi) \,|\, \varphi \in P \cup F \cup U\})$ l'hypergraphe associé au réseau PFU ;

2) dans le cas général : $O((|P \cup F| + 1) \cdot d^{1+w_{\mathcal{G}}(\preceq_{Sov})})$, avec $\mathcal{G} = (V, \{sc(\varphi) \,|\, \varphi \in P \cup F \cup \{U_0\}\})$ l'hypergraphe associé au réseau PFU dans lequel toutes les fonctions locales d'utilité sont fusionnées pour donner une unique fonction d'utilité U_0.

Exemple. Considérons la requête sur le problème des capteurs définie par la séquence $Sov = \max_{te} \oplus_{urte} \max_{rp} \oplus_{uecs,ec_A,ec_B}$. Cette séquence impose les contraintes $\{te\} \prec_{Sov} \{rte\} \prec_{Sov} \{rp\} \prec_{Sov} \{ecs, ec_A, ec_B\}$. Il est possible de montrer que l'ordre d'élimination $o : te \prec rte \prec rp \prec ecs \prec ec_A \prec ec_B$ a une largeur induite $w_{\mathcal{G}}(o) = 4$ et que cet ordre est optimal (la largeur induite contrainte vaut 4).

4.3. *Quelques sous-classes polynomiales du problème de réponse à une requête*

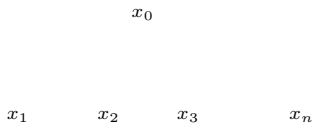
Le problème de décision associé à une requête PFU consiste à déterminer si $Ans(Q) \succeq \theta$ pour un certain seuil θ. Ce problème est PSPACE-complet (Pralet, 2006). La largeur induite contrainte nous permet toutefois de déduire certaines sous-classes polynomiales. Nous supposons par la suite que Ax^{SA} ou Ax^{SG} est satisfait.

Tout d'abord, si les réseaux PFU considérés sont de largeur induite bornée, alors les requêtes n'utilisant qu'un seul opérateur d'élimination sont polynomiales. Ce résultat évident peut être utilisé pour déduire la complexité polynomiale des POMDP possibilistes optimistes à horizon fini, qui n'utilisent en réalité que l'opérateur $\max$ comme opérateur d'élimination (leur largeur induite contrainte vaut 2, d'où une complexité cubique en la taille de l'espace d'état). Ce résultat ne s'applique par contre pas aux POMDP possibilistes pessimistes, dans lesquels on trouve une alternance entre opérations de minimisation et de maximisation.

La complexité reste également polynomiale lorsque les requêtes considérées sont toutes de largeur induite contrainte bornée. Ce résultat s'applique aux MDP à horizon fini, qu'ils soient probabilistes ou possibilistes, dont la largeur induite contrainte vaut 2.

Enfin, si l'hypergraphe associé à un réseau PFU (V, G, P, F, U) est un arbre et si la séquence Sov définissant la requête est "compatible" avec cet arbre, c'est-à-dire si chaque élimination op_S de Sov est telle que les chemins qui vont des variables de S aux feuilles de l'arbre contiennent soit des variables de S soit des variables éliminées à droite de op_S, alors répondre à la requête $(Sov, (V, G, P, F, U))$ peut se faire en temps et en espace polynomial.

Par contre, si la séquence d'élimination de variables n'est pas compatible avec l'arbre, le problème n'est plus nécessairement polynomial. La complexité d'un problème peut donc fortement dépendre de la séquence Sov considérée. Par exemple, pour le réseau $\mathcal{N}$ dont l'hypergraphe des fonctions locales est donné à la figure ci-contre, la requête $(\max_{x_0} \sum_{x_1,\ldots,x_n}, \mathcal{N})$ est polynomiale (largeur induite contrainte égale à 1) alors que la requête $(\sum_{x_1,\ldots,x_n} \max_{x_0}, \mathcal{N})$ est a priori beaucoup plus difficile (largeur induite contrainte égale à n).



5. Conclusion

Cet article a introduit un algorithme d'élimination de variables nommé **MVE**. Cet algorithme est capable de répondre à une requête sur un réseau PFU tout en utilisant les factorisations de quantités globales en fonctions locales pourvu qu'une des deux conditions de décomposabilité (axiome Ax^{SA} ou Ax^{SG}) soit satisfaite. L'utilisation de cet algorithme est résumée par le tableau 3, qui montre que dans le cas semi-anneau, son utilisation est plutôt naturelle, que dans le cas semi-groupe, elle requiert l'utilisation d'éléments appelés potentiels, et que dans le cas général, elle nécessite de combiner toutes les fonctions d'utilité locales en une fonction d'utilité globale.

L'algorithme **MVE** est ainsi un algorithme générique qui vient compléter le spectre des domaines d'applicabilité des algorithmes d'élimination de variables (qui jusqu'alors avaient surtout été étudiés dans le domaine mono-opérateur). Plus généralement, il appuie l'idée selon laquelle l'applicabilité de tel ou tel algorithme repose non pas sur l'aspect probabiliste ou non d'un formalisme (qui est important au niveau sémantique) mais sur les propriétés algébriques des opérateurs utilisés. Les deux grandes classes axiomatiques identifiées (Ax^{SA} et Ax^{SG}) confirment ce point car elles contiennent chacune des modèles probabilistes ou non probabilistes. Nous avons enfin montré que la complexité de l'algorithme **MVE** était fortement liée aux caractéristiques structurelles des problèmes considérés, qui dépendent à la fois de l'hypergraphe des fonctions locales manipulées et de la séquence d'éliminations de variables utilisée.

Ces caractéristiques structurelles pourraient par la suite être davantage étudiées, de manière à relâcher certaines contraintes sur l'ordre d'élimination, afin de pouvoir diminuer la largeur induite contrainte. Certaines éliminations utilisant des opérateurs différents peuvent en effet parfois être interverties, comme c'est le cas dans l'expression $\max_x \sum_y (\varphi_x \times \varphi_y) = \sum_y \max_x (\varphi_x \times \varphi_y)$. Ainsi, nous pourrions suivre une approche cherchant à révéler des libertés cachées dans l'ordre d'élimination, à la manière des travaux développés dans (Pralet *et al.*, 2006b; Pralet *et al.*, 2006a; Pralet, 2006).

6. Bibliographie

Aji S., McEliece R., « The Generalized Distributive Law », *IEEE Transactions on Information Theory*, vol. 46, n° 2, p. 325-343, 2000.

Arnborg S., « Efficient Algorithms for Combinatorial Problems on Graphs with Bounded Decomposability - A Survey », *BIT*, vol. 25, p. 2-23, 1985.

Bertelé U., Brioschi F., *Nonserial Dynamic Programming*, Academic Press, 1972.

Bordeaux L., Monfroy 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, 2002.

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, p. 291-296, 2003.

Dechter R., « Bucket Elimination : a Unifying Framework for Reasoning », *Artificial Intelligence*, vol. 113, n° 1-2, p. 41-85, 1999.

Dechter R., *Constraint Processing*, Morgan Kaufmann, 2003.

Dechter R., Fattah Y. E., « Topological Parameters for Time-Space Tradeoff », *Artificial Intelligence*, vol. 125, n° 1-2, p. 93-118, 2001.

Dubois D., Prade H., « Possibility Theory as a Basis for Qualitative Decision Theory », *Proc. of the 14th International Joint Conference on Artificial Intelligence (IJCAI-95)*, Montréal, Canada, p. 1925-1930, 1995.

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.

Frydenberg M., « The Chain Graph Markov Property », *Scandinavian Journal of Statistics*, vol. 17, p. 333-353, 1990.

Garcia L., Sabbadin R., « Possibilistic Influence Diagrams », *Proc. of the 17th European Conference on Artificial Intelligence (ECAI-06)*, Riva del Garda, Italy, p. 372-376, 2006.

Giang P., Shenoy P., « A Qualitative Linear Utility Theory for Spohn's Theory of Epistemic Beliefs », *Proc. of the 16th International Conference on Uncertainty in Artificial Intelligence (UAI-00)*, Stanford, California, USA, p. 220-229, 2000.

Giang P., Shenoy P., « Two Axiomatic Approaches to Decision Making Using Possibility Theory », *European Journal of Operational Research*, vol. 162, n° 2, p. 450-467, 2005.

Goldman R., Boddy M., « Expressive Planning and Explicit Knowledge », *Proc. of the 3rd International Conference on Artificial Intelligence Planning Systems (AIPS-96)*, Edinburgh, Scotland, p. 110-117, 1996.

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.

Jensen F., Jensen F., Dittmer S., « From Influence Diagrams to Junction Trees », *Proc. of the 10th International Conference on Uncertainty in Artificial Intelligence (UAI-94)*, Seattle, WA, USA, p. 367-373, 1994.

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

Monahan G., « A Survey of Partially Observable Markov Decision Processes : Theory, Models, and Algorithms », *Management Science*, vol. 28, n° 1, p. 1-16, 1982.

Ndilikilikesha P., « Potential Influence Diagrams », *International Journal of Approximated Reasoning*, vol. 10, p. 251-285, 1994.

Papadimitriou C., *Computational Complexity*, Addison-Wesley Publishing Company, 1994.

Park J., Darwiche A., « Solving MAP Exactly using Systematic Search », *Proceedings of the 19th Annual Conference on Uncertainty in Artificial Intelligence (UAI-03)*, San Francisco, CA, p. 459-468, 2003.

Park J., Darwiche A., « Complexity Results and Approximation Strategies for MAP Explanations », *Journal of Artificial Intelligence Research*, vol. 21, p. 101-133, 2004.

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

Pralet C., A Generic Algebraic Framework for Representing and Solving Sequential Decision Making Problems with Uncertainties, Feasibilities, and Utilities, PhD thesis, Ecole Nationale Supérieure de l'Aéronautique et de l'Espace, Toulouse, France, 2006.

Pralet C., Schiex T., Verfaillie G., « Decomposition of Multi-Operator Queries on Semiring-based Graphical Models », *Proc. of the 12th International Conference on Principles and Practice of Constraint Programming (CP-06)*, Nantes, France, p. 437-452, 2006a.

Pralet C., Schiex T., Verfaillie G., « From Influence Diagrams to Multioperator Cluster DAGs », *Proc. of the 22nd International Conference on Uncertainty in Artificial Intelligence (UAI-06)*, Cambridge, MA, USA, 2006b.

Pralet C., Verfaillie G., Schiex T., « Decision with Uncertainties, Feasibilities, and Utilities : Towards a Unified Algebraic Framework », *Proc. of the 17th European Conference on Artificial Intelligence (ECAI-06)*, Riva del Garda, Italy, p. 427-431, 2006c.

Pralet C., Verfaillie G., Schiex T., « Un Cadre Graphique et Algébrique pour les Problèmes de Décision incluant Incertitudes, Faisabilités et Utilités », *Revue d'Intelligence Artificielle*, vol. 21, n° 3, p. 419-448, 2007.

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

Robertson N., Seymour P., « Graph Minors II : Algorithmic Aspects of Treewidth », *Journal of Algorithms*, vol. 7, p. 309-322, 1986.

Sabbadin R., « A Possibilistic Model for Qualitative Sequential Decision Problems under Uncertainty in Partially Observable Environments », *Proc. of the 15th International Conference on Uncertainty in Artificial Intelligence (UAI-99)*, Stockholm, Sweden, p. 567-574, 1999.

Schiex T., Fargier H., Verfaillie G., « Valued Constraint Satisfaction Problems : Hard and Easy Problems », *Proc. of the 14th International Joint Conference on Artificial Intelligence (IJCAI-95)*, Montréal, Canada, p. 631-637, 1995.

Shafer G., *A Mathematical Theory of Evidence*, Princeton University Press, 1976.

Shenoy P., « Valuation-based Systems for Discrete Optimization », *Proc. of the 6th International Conference on Uncertainty in Artificial Intelligence (UAI-90)*, Cambridge, MA, USA, p. 385-400, 1990.

Shenoy P., « Using Possibility Theory in Expert Systems », *Fuzzy Sets and Systems*, vol. 52, n° 2, p. 129-142, 1992.

Shenoy P. P., Shafer G., « Axioms for Probability and Belief-Function Propagation », *Proc. of the 4th International Conference on Uncertainty in Artificial Intelligence (UAI-88)*, Minneapolis, MN, USA, p. 169-198, 1988.

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, p. 149-158, 1990.

von Neumann J., Morgenstern O., *Theory of Games and Economic Behaviour*, Princeton University Press, 1944.

Wilson N., « An Order of Magnitude Calculus », *Proc. of the 11th International Conference on Uncertainty in Artificial Intelligence (UAI-95)*, Montréal, Canada, p. 548-555, 1995.