Les intégrales de Choquet basées sur les mesures de nécessité pour la prise de décision séquentielle sous incertitude

N. Ben Amor[1] *H. Fargier*[2] *W. Guezguez*[1]
[1] LARODEC, Institut Supérieur de Gestion Tunis, Tunisie
[2] IRIT-CNRS, UMR 5505 Université de Toulouse, France
nahla.benamor@gmx.fr, fargier@irit.fr, widedguezguez@gmail.com

Résumé :
La théorie des possibilités se pose en alternative à la théorie de l'utilité espérées lorsque l'incertitude sur les données ne peut pas être quantifiée de manière probabiliste. En présence d'informations hétérogènes, comme c'est le cas lorsque les connaissances sur l'état du monde sont modélisées par une distribution de possibilité alors que les degrés d'utilité sont numériques et compensatoires, il faut évaluer chaque décision en utilisant sa valeur de Choquet basée sur la mesure de nécessité. Dans cet article, nous étudions comment ce critère peut être utilisé en décision séquentielle, plus précisément dans les arbres de décision. Nous montrons que, à l'instar des intégrales de Choquet générales, les intégrales de Choquet possibilistes ne satisfont pas la propriété de monotonie sur laquelle s'appuient les algorithmes de programmation dynamique classiquement associés aux arbres de décision. Ensuite, nous proposons un algorithme de Branch and Bound basé sur une évaluation optimiste de la valeur de Choquet des arbres de décision possibilistes.

Mots-clés :
Arbre de décision, théorie des possibilités, intégrale de Choquet, programmation dynamique.

Abstract:
Possibility theory is a natural one to consider when information about uncertainty cannot be quantified in probabilistic way. In presence of heterogeneous information, i.e. when the knowledge about the state of the world is modeled by a possibility distribution while the utility degrees are numerical and compensatory, one should evaluate each decision on the basis of its Necessity-based Choquet value. The present paper studies the use of this criterion in the context of sequential decision trees. We show that it does not satisfy the monotonicity property on which rely the dynamic programming algorithms classically associated to decision trees. Then, we propose a Branch and Bound algorithm using an optimistic evaluation of the Choquet value of possibilistic decision trees.

Keywords:
Decision tree, possibility theory, Choquet integral, dynamic programming

1 Introduction

La prise de décision sous incertitude est l'un des principaux domaines de recherche en théorie de décision, du fait de ses nombreuses applications (par exemple médecine, contrôle de robot, décision stratégique, jeux...). Dans ces problèmes, la conséquence d'une décision dépend d'événements incertains. Lorsqu'on peut supposer qu'une probabilité est précisément connue pour chaque événement, on parle de décision sous risque ; dans ce contexte, chaque décision peut être représentée par une "loterie probabiliste", c'est à dire une distribution de probabilité sur les conséquences possibles de la décision. Ce modèle s'étend aux problèmes de décision dit séquentiels où une séquence de décisions doit être prise conditionnellement à des états observés - on parle de "stratégie" : une stratégie est simplement une fonction affectant une décision (c'est à dire une loterie sur les états suivants) à chaque état atteignable. La loterie composée définie par la stratégie peut alors être réduite, évaluée, et comparée à d'autres. Le critère fondamental de comparaison des loteries probabilistes, et donc de stratégies, est celui de l'utilité espérée axiomatisée par Von Neumann et Morgenstern [8]. Ce critère suppose que les préférences du décideur sont capturées par une fonction d'utilité numérique. L'évaluation d'une loterie est alors effectuée par le calcul de son utilité espérée. Lorsque le problème est séquentiel, le nombre de stratégies possibles augmente exponentiellement, mais l'utilité espérée possède une propriété de monotonie précieuse, qui permet d'éviter une exploration exhaustive de l'espace des stratégies puisqu'elle permet de construire une stratégie optimale par application d'un algorithme polynomial de programmation dynamique.

Lorsqu'il n'est pas possible d'avoir recours à la théorie des probabilités, faute de pouvoir quan-

tifier précisément les informations incertaines, mais que l'on possède une information de nature plus ordinale, la théorie de décision possibiliste représente une alternative très attractive [3][1]. Dans ce contexte, deux critères qualitatifs possibilistes ont été proposés et axiomatisés, l'un prudent (fondé sur une mesure de nécessité), et l'autre plus aventureux, fondé sur une mesure de possibilité. Ces deux critères se basent uniquement sur une échelle finie ordinale non compensatoire pour l'évaluation de l'utilité et de la plausibilité. Ils sont compatibles avec la propriété de monotonie sus citée, qui a abouti au développement des modèles qualitatifs sophistiqués pour la prise de décision possibiliste séquentielle comme les processus de décision markoviens possibilistes [9] [12], les arbres de décision possibilistes ordinaux [4].

En présence d'informations hétérogènes, en particulier quand la connaissance à propos de l'état du monde est possibiliste alors que les degrés d'utilité sont numériques et compensatoires, les modèles précédents ne s'appliquent plus. Gilboa et Schmeidler [13], [5] proposent alors d'étendre l'utilité espérée aux modèles non Bayésiens, par l'utilisation d'intégrales de Choquet. Ce modèle permet une modélisation de l'incertitude par une capacité quelconque (e.g. une mesure de nécessité). Néanmoins, son utilisation n'est pas aisée : il ne satisfait pas le principe de la monotonie dans le cas général. Par conséquent, l'optimalité de la solution produite par la programmation dynamique n'est pas garantie. La question qui se pose est alors de savoir si l'intégrale de Choquet possibiliste satisfait le principe de monotonie, et si pas, quel algorithme peut être utilisé pour le calcul d'une stratégie optimale.

Dans cet article, nous montrons que les intégrales de Choquet basées sur des mesure de nécessité ne satisfont pas la propriété de la monotonie et nous proposons un algorithme de Branch and Bound basé sur une évaluation optimiste de la valeur de Choquet des arbres de décision possibilistes [1]. Les notions de base de la théorie des possibilités sont rappelées dans la section 2, les arbres de décision possibilistes sont présentés dans la section 3. La quatrième section est réservée à notre algorithme de résolution. Enfin, la section 5 présente des expérimentations des algorithmes proposés.

2 Notions de base

Cette section donne un bref aperçu de la théorie des possibilités (pour plus de détails voir [2]) et de la théorie de la décision possibiliste [1] [3].

Soit x une variable dont la valeur, mal connue appartient à un domaine D (fini ou non). La connaissance sur les valeurs plus ou moins possibles de x est décrite par une distribution de possibilité $\pi : D \rightarrow [0,1]$; $\pi(\omega) = 1$ signifie que la valeur de ω est totalement possible et $\pi(\omega) = 0$ signifie que ω est impossible. On peut définir deux mesures duales pour chaque événement $A \subseteq D$: la mesure de possibilité qui représente à quel niveau la proposition $x \in A$ est cohérente avec les connaissances représentées par π et la mesure de nécessité qui correspond au degré de certitude associé à cette proposition :

$$\Pi(x \in A) = sup_{v \in A} \pi(v) \quad (1)$$

$$N(x \in A) = 1 - \Pi(x \notin A) = inf_{v \notin A} 1 - \pi(v) \quad (2)$$

Soient $x_1, \ldots, x_n$, n variables non interactives (logiquement indépendantes) ; soit π_i la distribution de possibilité relative à x_i. La distribution jointe $\pi(.)$ est définie par :

$$\pi(v_1, \ldots, v_n) = \pi_1(v_1) \otimes \cdots \otimes \pi_n(v_n) \quad (3)$$

La particularité de l'échelle possibiliste est qu'elle peut être interprétée de deux façons. La première correspond à une interprétation ordinale des degrés de possibilité : l'opérateur *min* est alors utilisé pour combiner les distributions

[1] Cet article étend un version préliminaire [7] en détaillant l'algorithme de programmation dynamique et présentant une analyse expérimentale qui montre son intérêt.

($\otimes = \min$). La deuxième interprétation est numérique : dans ce cas $\otimes$ est l'opérateur produit ($\otimes = *$).

Un acte peut être vu comme une distribution de possibilité sur un ensemble d'états. Modélisant l'attractivité de chaque état pour le décideur par une fonction d'utilité lui attribuant une valeur de l'ensemble $U = \{u_1, \ldots, u_n\} \subseteq \mathbb{R}$ (nous supposons sans perte de généralité que $u_1 \leq \cdots \leq u_n$), on peut directement modéliser cet acte par une distribution de possibilité sur U, appelée aussi loterie possibiliste simple et notée par $(\lambda_1/u_1, \ldots, \lambda_n/u_n)$ où λ_i est la possibilité qu'un acte entraîne une utilité u_i[2] (voir [3], [1]). On note $\mathcal{L}$ l'ensemble des loteries simples que l'on peut construire sur U et $\mathcal{L}_c \subseteq \mathcal{L}$ le sous ensemble de loteries certaines dans $\mathcal{L}$: $\mathcal{L}_c = \{\pi$ tel que $\exists u_i, \pi(u_i) = 1$ et $\forall u_j \neq u_i, \pi(u_j) = 0\}$. On dit qu'une loterie $L^i \in \mathcal{L}$ *domine* une loterie $L^j \in \mathcal{L}$ (noté $L^i \succeq_S L^j$) ssi :

$$\forall u_i, N(L^i \geq u_i) \geq N(L^j \geq u_i) \qquad (4)$$

Une loterie possibiliste composée (relative à $\mathcal{L}$) notée par $(\lambda_1/L^1, \ldots, \lambda_m/L^m)$ est une distribution de possibilité sur un sous ensemble de $\mathcal{L}$. On peut la réduire à une loterie simple comme suit : la possibilité $\pi_{i,j}$ de recevoir le degré d'utilité u_j à partir d'une des sous loteries L^i dépend de la possibilité λ_i d'avoir L^i et de la possibilité λ_j^i d'avoir u_j de L^i ; autrement dit $\pi_{i,j} = \lambda_j \otimes \lambda_j^i$. Plus généralement, la possibilité de recevoir u_j à partir de la loterie composée $(\lambda_1/L^1, ..., \lambda_m/L^m)$ est simplement le *max* des $\pi_{i,j}$ sur toutes les loteries simples $L^i \in \mathcal{L}$. [3][1] ont proposé de réduire $(\lambda_1/L^1, ..., \lambda_m/L^m)$ à une loterie simple définie par :

$$\lambda_1 \otimes L^1 \oplus ... \oplus \lambda_m \otimes L^m$$

$$= (max_{j=1,m} \lambda_1^j \otimes \lambda_j/u_1, \ldots, max_{j=1,m} \lambda_n^j \otimes \lambda_j/u_n) \qquad (5)$$

Cette loterie est considérée comme un équivalent à $(\lambda_1/L^1, \ldots, \lambda_m/L^m)$ et elle est appelée la *réduction* de la loterie composée.

[2]Pour simplifier la notation, on ne renseignera dans le vecteur que les u_i tels que $\pi(u_i) > 0$

Sous l'hypothèse que l'échelle d'utilité et l'échelle d'évaluation des degrés de possibilité sont commensurables et purement ordinales, Dubois et Prade [3] ont proposé un degré d'utilité possibiliste pessimiste (U_{pes}) et un degré d'utilité possibilité optimistes (U_{opt}) pour l'évaluation de loteries possibilistes $L \in \mathcal{L}$ comme suit :

$$U_{pes}(L) = max_{i=1,n} \min(u_i, N(L \geq u_i)) \quad (6)$$

$$U_{opt}(L) = max_{i=1,n} \min(u_i, \Pi(L \geq u_i)) \quad (7)$$

où $N(L \geq u_i) = 1 - \Pi(L < u_i) = 1 - max_{j=1,i-1}\lambda_j$ et $\Pi(L \geq u_i) = max_{j=i,n}\lambda_j$ sont les degrés de nécessité et de possibilité que L atteigne au moins la valeur de l'utilité u_i.

Ces deux degrés sont des cas particuliers de l'intégrale de Sugeno exprimée par :

$$S_\mu(L) = max_{i=1,n} \min(u_i, \mu(L \geq u_i)) \quad (8)$$

où μ est une capacité, c'est à dire une fonction d'ensemble telle que $\mu(\emptyset) = 0, \mu(\Omega) = 1, A \subseteq B \Rightarrow \mu(A) \leq \mu(B)$.

Notons que U_{pes} est retrouvée quand μ est une mesure de nécessité. Sous la même hypothèse de commensurabilité mais en supposant que les utilités ont une interprétation plus riche, cardinale, on évalue l'utilité de L sur la base d'une intégrale de Choquet :

$$Ch_\mu(L) = \Sigma_{i=n,1}(u_i - u_{i-1}) \,.\, \mu(L \geq u_i) \quad (9)$$

Si μ est une mesure de probabilité alors $Ch_\mu(L)$ est simplement l'utilité espérée de L. Dans la suite, nous sommes intéressées au cas particulier d'intégrales de Choquet possibilistes pessimistes, autrement dit nous allons considérer que μ est une mesure de nécessité. Dans ce cas, l'équation (9) s'écrit comme suit :

$$Ch_N(L) = \Sigma_{i=n,1}(u_i - u_{i-1}).N(L \geq u_i). \qquad (10)$$

3 Les arbres de décision possibilistes

Les arbres de décision [10] sont des représentations graphiques de problèmes de décision séquentiels sous l'hypothèse de l'observation totale (on suppose une fois qu'une décision a été

exécutée, son résultat est connu et observé). Un arbre de décision (voir comme exemple la figure 1) est un arbre $GT = (\mathcal{N}, \mathcal{E})$. L'ensemble de noeuds $\mathcal{N}$ contient trois types de noeuds :

- $\mathcal{D} = \{D_0, \ldots, D_m\}$ est l'ensemble de noeuds décision (représentés par des rectangles). La numérotation des noeuds est supposée conforme avec l'ordre temporel, à savoir si D_i est un descendant de D_j alors $i > j$. La racine de l'arbre est nécessairement un noeud d décision, noté par D_0.
- $\mathcal{LN} = \{LN_1, \ldots, LN_k\}$ est l'ensemble de feuilles, appelés aussi des feuilles d'utilités $\forall LN_i \in \mathcal{LN}$, $u(LN_i)$ est l'utilité d'être éventuellement dans le noeud LN_i.
- $\mathcal{C} = \{C_1, \ldots, C_n\}$ est l'ensemble des noeuds chance (représentés par des cercles). Ils représentent les actions possibles (et leurs fils les résultats possibles de ces actions).

$\forall X_i \in \mathcal{N}$, $Succ(X_i) \subseteq \mathcal{N}$ dénote l'ensemble de ses fils. $\forall D_i \in \mathcal{D}, Succ(D_i) \subseteq \mathcal{C}$, $Succ(D_i)$ est l'ensemble des actions qui peuvent être décidées quand D_i est observé. $\forall C_i \in \mathcal{C}, Succ(C_i) \subseteq \mathcal{LN} \cup \mathcal{D}$, $Succ(C_i)$ est en effet l'ensemble des résultats possibles de l'action C_i : soit un noeud feuille est observé (et son utilité est engrangée), soit un noeud décision est observé (et une nouvelle action doit être exécutée). Dans les arbres de décision classiques probabilistes, l'incertitude portant sur le résultat de chaque C_i est représentée par une distribution de probabilité sur $Succ(C_i)$. Ici, nous utilisons évidement un étiquetage possibiliste : pour tout $C_i \in \mathcal{C}$, la connaissance sur les conséquences C_i est représentée par une distribution de possibilité : $\pi_i(Y), \forall Y \in Succ(C_i)$ est la possibilité que Y soit la conséquence de l'action C_i . Le problème est alors de construire une stratégie qui choisit une action (à savoir un noeud chance) par noeud décision. Formellement, on définit une stratégie comme une fonction δ de $\mathcal{D}$ dans $\mathcal{C} \cup \{\perp\}$. $\delta(D_i)$ est l'action à exécuter quand un noeud de décision D_i est observé. $\delta(D_i) = \perp$ signifie qu'aucune action n'a été sélectionnée pour D_i (car aucun D_i ne peut être atteint ou la stratégie est partiellement définie).

Les stratégies admissibles doivent être :
valides : $\forall D_i, \delta(D_i) \in Succ(D_i) \cup \{\perp\}$
complètes : (i) $\delta(D_0) \neq \perp$ et (ii) $\forall D_i$ t. q $\delta(D_i) \neq \perp, \forall X_j \in Succ(\delta(D_i))$, ou bien $\delta(X_j) \neq \perp$ ou $X_j \in \mathcal{LN}$

Soit Δ l'ensemble des stratégies valides et complètes qui peuvent être construites à partir de l'arbre GT. Toute stratégie de Δ peut être vue comme un sous arbre connecté de l'arbre de décision ; ses arcs sont de la forme $(D_i, \delta(D_i))$.

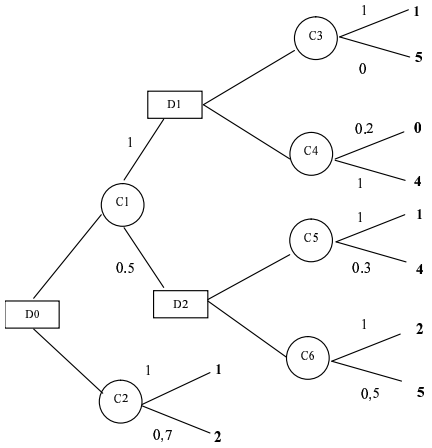

Figure 1 – Exemple d'un arbre de décision possibiliste avec $\mathcal{C} = \{C_1, C_2, C_3, C_4, C_5, C_6\}$, $\mathcal{D} = \{D_0, D_1, D_2\}$ et $\mathcal{LN} = U = \{0, 1, 2, 3, 4, 5\}$.

Dans cet article, nous interprétons les degrés d'utilités de manière numérique, compensatoire. Nous nous sommes donc intéressés à l'obtention de stratégies de Δ maximisant le critère basé sur Choquet possibiliste pessimiste, basé sur un mesure de nécessité (voir [11] pour une justification axiomatique de ce critère ; nous avons laissé de coté le cas optimiste, trop aventureux). La valeur de Choquet d'une stratégie peut être déterminée grâce à la notion de réduction de loterie. Les noeuds feuilles ln dans $\mathcal{LN}$ étant étiquetés par des degrés d'utilités, ou, de manière équivalente des loteries certaines de $\mathcal{L}_c$, un noeud chance peut être vu comme une loterie composée. Pour évaluer une stratégie on va donc la réduire en une loterie simple équivalente, dont on calculera l'utilité selon Choquet. Formellement, la composition de loteries sera appliquée à partir des feuilles d'une stratégie à sa racine, selon la définition récursive suivante :

$$\forall X_i \in \mathcal{N}, L(X_i, \delta) = \begin{cases} L(\delta(X_i), \delta) \ \ si \ \ X_i \in \mathcal{D} \\ Max_{X_j \in Succ(X_i)} \pi_i(X_j) \otimes L(X_j, \delta) \ si \ X_i \in \mathcal{C} \\ (1/u(X_i)) \ \ si \ \ X_i \in \mathcal{LN} \end{cases} \tag{11}$$

où $\otimes = *$ ou $\otimes = min$ selon l'interprétation des degrés de possibilité. L'équation (11) est l'adaptation aux stratégies de la réduction de loterie (équation (5)). On peut alors calculer $L(\delta) = L(D_0, \delta)$: $L(\delta)(u_i)$ est simplement la possibilité d'avoir u_i quand δ est appliquée à partir de D_0. Par conséquent :

$$Ch_N(\delta) = Ch_N(L(D_0, \delta)). \tag{12}$$

4 Recherche d'une stratégie Choquet optimale dans les arbres de décision possibilistes

Étant donné un arbre de décision possibiliste représentant un ensemble de stratégies admissibles $\Delta = \{\delta_1 \ldots \delta_n\}$, le problème est d'obtenir une stratégie δ^* tel que $\forall \delta \in \Delta$, $Ch_N(\delta^*) \geq Ch_N(\delta)$. Malheureusement, la recherche des stratégies optimales via une énumération exhaustive de Δ est une tâche exponentielle. Par exemple, dans un arbre de décision comportant n noeuds de décision binaires, le nombre de stratégies potentielles est de l'ordre de $O(2^{\sqrt{n}})$.

Pour les arbres de décision probabilistes où le but est de maximiser l'utilité espérée, une stratégie optimale peut être calculée en temps polynomial (en fonction de la taille de l'arbre) via un algorithme de programmation dynamique. Cet algorithme construit la meilleure stratégie en optimisant les décisions depuis les derniers noeuds décision de l'arbre jusqu'à sa racine. En ce qui concerne les arbres de décision possibilistes, Garcia et Sabbadin [4] ont montré que la programmation dynamique peut être utilisée, permettant l'obtention en temps polynomial une stratégie maximisant U_{pes}. En effet, U_{pes} satisfait la propriété clé de monotonie :

Soit V un critère de décision, V est dit monotone ssi quelque soient les loteries L^i, L^j et L^k, quelque soit la distribution de possibilité normalisée (α,β) : $V(L^i) \geq V(L^j) \Rightarrow V((\alpha \otimes L^i) \oplus (\beta \otimes L^k)) \geq V((\alpha \otimes L^j) \oplus (\beta \otimes L^k))$.

Cette propriété stipule que la combinaison L^i (resp L^j) avec une troisième loterie L^k préserve l'ordre induit par V entre L^i et L^j. Elle permet de décider en faveur de L^i avant de considérer la décision composée ; d'où la possibilité d'un choix depuis les décisions basses de l'arbre jusqu'à la racine : c'est le principe de la programmation dynamique. Il est facile d'adapter cet algorithme (polynomial) au cas des intégrales de Choquet possibilistes (cf. algorithme 1).

Algorithm 1: Fonction ProgDyn

Données: un noeud X, une stratégie δ
Résultat: L^X *% $L^X[u_i]$ est le degré de possibilité d'avoir l'utilité u_i*

```
début
    pour i ∈ {1,..,n} faire L^X[u_i] ← 0
    si X ∈ LN alors
        L^X[u(X)] ← 1
    si X ∈ C alors
        pour chaque Y ∈ Succ(X) faire
            L^Y ← ProgDyn(Y, δ) pour
            i ∈ {1,..,n} faire
                L^X[u_i] ←
                max(L^X[u_i], π_X(Y) ⊗ L^Y[u_i])
    si X ∈ D alors
        si δ(X) ≠ ⊥ alors L^X ← ProgDyn(δ(X), δ)
        sinon
            si |Succ(X)| = 1 alors
                L^X ← ProgDyn(δ(Succ(X)), δ)
            sinon max ← 0 pour chaque
            Y ∈ Succ(X) ∩ N_δ faire
                L^Y ← ProgDyn(Y, δ) si
                Ch_N(L^Y) > max alors
                    max ← Ch_N(L^Y)
                    Y* ← Y
                δ(X) ← Y*
    retourner L^X
fin
```

La question est de savoir si cet algorithme se comporte dans notre cadre comme pour le cas probabilistes ou pour le cas purement ordinal ? On sait que la monotonie n'est pas toujours satisfaite par certaines intégrales de Choquet, mais que ce n'est pas le cas général (en effet, certaines intégrales de Choquet, comme celles représentant le modèle de l'utilité dépendant du rang, ne satisfont pas cette condition [6]). Nous

montrons dans les contre-exemples suivants que ce n'est pas le cas non plus pour celles qui nous intéressent, à savoir les intégrales de Choquet basées sur les mesures de nécessité comme le montrent les deux contre-exemples suivants.

Contre-exemple 1. *[Pour $\otimes = *$]*
Soit $L^1 =< 0.1/1, 1/2, 0/3 >$ *et* $L^2 =< 0.9/1, 0/2, 1/3 >$. $Ch_N(L^1) = 1.9$ *et* $Ch_N(L^2) = 1.2$, *donc* $L^1 \succ L^2$. *Soient* $L^3 =< 1/1, 0.1/2, 0/3 >$, $\alpha = 1$, $\beta = 0.9$, $L^{c1} = (\alpha \otimes L^1) \oplus (\beta \otimes L^3)$ *et* $L^{c2} = (\alpha \otimes L^2) \oplus (\beta \otimes L^3)$. *D'après l'équation (5) :* $L^{c1} =< 0.9/1, 1/2, 0/3 >$ *et* $L^{c2} =< 0.9/1, 0.09/2, 1/3 >$. *D'où* $Ch_N(L^{c1}) = 1.1 < Ch_N(L^{c2}) = 1.2$, *ce qui contredit la propriété de monotonie.*

Contre-exemple 2. *[Pour $\otimes = \min$]*
Soit $L^1 =< 0.2/0, 0.5/0.51, 1/1 >$ *et* $L^2 =< 0.1/0, 0.6/0.5, 1/1 >$. $Ch_N(L^1) = 0.653$ *et* $Ch_N(L^2) = 0.650$, *donc* $L^1 \succ L^2$. *Soient* $L^3 =< 0.01/0, 1/1 >$, $\alpha = 0.55$, $\beta = 1$, $L^{c1} = (\alpha \otimes L^1) \oplus (\beta \otimes L^3)$, $L^{c2} = (\alpha \otimes L^2) \oplus (\beta \otimes L^3)$. *D'après l'équation (5) :* $L^{c1} =< 0.2/0, 0.5/0.51, 1/1 >$ *et* $L^{c2} =< 0.1/0, 0.55/0.5, 1/1 >$. *D'où* $Ch_N(L^{c1}) = 0.653 < Ch_N(L^{c2}) = 0.675$, *ce qui contredit la monotonie.*

La propriété de monotonie n'étant pas satisfaite par l'intégrale de Choquet basée sur la mesure de nécessité, l'utilisation de l'algorithme de programmation dynamique peut conduire à une stratégie sous optimale. Comme alternative, nous avons choisi de procéder par une énumération implicite des stratégies via un algorithme de Branch and Bound. Ce type d'algorithme a déjà été utilisé pour l'optimisation du critère d'utilité dépendant du rang (cf. [6]). L'algorithme *BB* prend en argument une stratégie partielle δ et une borne supérieure de la meilleure valeur de Choquet que cette stratégie peut atteindre et il retourne la valeur Ch_N^{opt} de la meilleure stratégie générée (δ^{opt}). Pour l'initialisation de δ^{opt}, une première stratégie est produite, par programmation dynamique. En effet, bien que cet algorithme ne produit pas nécessairement la stratégie optimale, il peut produire une borne intéressante pour celle ci. Le premier appel du BB se fait avec la stratégie vide ($\forall D_i$, $\delta(D_i) = \perp$). A chaque étape, la stratégie partielle courante, δ est développée par le choix d'une action pour un noeud décision non affecté. Quand plusieurs noeuds de décision sont candidats pour être développés, celui ayant le rang minimal aura la priorité. La récursion s'arrête lorsque la stratégie courante est complète : dans ce cas, δ^{opt} et Ch_N^{opt} sont éventuellement mis à jour.

Algorithm 2: Fonction BB

Données: Une stratégie δ, sa valeur de Choquet, Ch_N^{δ}
Résultat: Ch_N^{opt}
début
  si $\delta = \emptyset$ **alors** $\mathcal{D}_{pend} \leftarrow \{D_1\}$ **sinon**
    $D_{pend} \leftarrow \{D_i \in \mathcal{D}$ t.q. $\delta(D_i) = \perp$ et $\exists D_j, \delta(D_j) \neq \perp$ et $D_i \in Succ(\delta(D_j))\}$
  si $\mathcal{D}_{pend} = \emptyset$ *(% δ est une stratégie complète)* **alors**
    si $Ch_N^{\delta} > Ch_N^{opt}$ **alors** $\delta^{opt} \leftarrow \delta$ **retourner** Ch_N^{δ}
  sinon
    $D_{next} \leftarrow arg\ min_{D_i \in D_{pend}}\ i$ **pour chaque** $C_i \in Succ(D_{next})$ **faire**
      $\delta((D_{next}) \leftarrow C_i)$
      $Eval \leftarrow Ch_N(Loterie(D_0, \delta))$ **si** $Eval > Ch_N^{opt}$ **alors**
        $Ch_N^{opt} \leftarrow \max(Ch_N^{opt}, \delta, Eval)$
  retourner Ch_N^{opt}
fin

Le calcul d'une borne supérieure de la valeur de Choquet pour la branche courante fait appel à la fonction $Loterie(\delta)$, qui retourne une loterie surmontant toutes les stratégies complètes compatibles avec δ. On peut montrer que :

$$Si\ L \succeq_S L'\ alors\ Ch_N(L) \geq Ch_N(L') \quad (13)$$

Par conséquent $Ch_N(Loterie(\delta))$ est une borne supérieure du meilleur développement de δ. Si $Ch_N(Loterie(\delta)) \leq Ch_N^{opt}$, l'algorithme backtraque chronologiquement, revenant sur le dernier choix d'action effectué. Remarquons que si δ est complète, $Loterie(\delta)$ retourne $L(D_0, \delta)$; la borne supérieure est égale à la valeur de Choquet quand la stratégie est complète. La fonction $Loterie$ prend en entrée une stratégie partielle. Il procède rétroactivement, des feuilles vers la racine de l'arbre. Une loterie simple $< 1/u(LN_i) >$ est associée à chaque feuille. A chaque noeud d'incertitude C_i, la procédure effectue la composition des loteries de $Succ(C_i)$ conformément à l'équation

Algorithm 3: Loterie

Données: un noeud X, une stratégie δ
Résultat: L^X *% $L^X[u_i]$ est le degré de possibilité d'avoir l'utilité u_i*

```
début
    pour i ∈ {1,..,n} faire L^X[u_i] ← 0 si X ∈ LN
    alors L^X[u(X)] ← 1 si X ∈ C alors
        pour chaque Y ∈ Succ(X) faire
            L^Y ← Loterie(Y, δ) pour i ∈ {1,..,n}
            faire
                L^X[u_i] ←
                max(L^X[u_i], π_X(Y) ⊗ L^Y[u_i])
    si X ∈ D alors
        si δ(X) ≠ ⊥ alors L^X ← Loterie(δ(X), δ)
        sinon
            si |Succ(X)| = 1 alors
                L^X ← Loterie(δ(Succ(X)), δ)
            sinon
                pour chaque Y ∈ Succ(X) ∩ N_δ
                faire
                    L^Y ← Loterie(Y, δ)
                    pour i ∈ {1,..,n} faire
                        G^c_Y[u_i] ←
                        1 − max_{u_j<u_i} L^Y[u_j]
                % Calculer l'enveloppe supérieure
                des fonctions cumulatives) pour
                i ∈ {1,..,n} faire
                    G^c[u_i] ←
                    max_{Y∈Succ(X)∩N_δ} G^c_Y[u_i]
                % Calculer Rev(G^c) L^X[u_n] ← 1
                pour i ∈ {n−1,..,1} faire
                    L^X[u_i] ← 1 − G^c[u_{i+1}]
    retourner L^X
fin
```

(11). A chaque noeud de décision D_i la procédure rend une loterie qui domine toutes les loteries de $Succ(D_i)$. Formellement, pour toute loterie $L \in \mathcal{L}$, G^c_L est la *fonction décumulative possibiliste* de L exprimée par :

$$G^c_L(u) = N(L \geq u) \quad \forall u \in U \qquad (14)$$

Étant donné un ensemble $\mathcal{G} = \{G^c_{L^1}, \ldots, G^c_{L^k}\}$ de fonctions décumulatives, *l'enveloppe supérieure* de $\mathcal{G}$ est la fonction décumulative $G^c_{\mathcal{G}}$:

$$G^c_{\mathcal{G}}(u) = max_{G^c_{L^i} \in \mathcal{G}} G^c_{L^i}(u) \ \ \forall u \in U \qquad (15)$$

Étant donnée une fonction décumulative G^c sur U, $Rev(G^c)$ est la loterie définie par :

$$Rev(G^c)(u_i) = \begin{cases} 1 - G^c(u_{i+1}) & si \ \ i \in \{1, \ldots, n-1\} \\ 1 & si \ \ i = n \end{cases} \qquad (16)$$

Il est facile de démontrer que la fonction décumulative possibiliste associée à la loterie $Rev(G^c)$ est égale à G^c [3]. Par conséquent, pour tout ensemble $\mathcal{L}$ de loteries simples dont l'ensemble de de fonctions décumulatives respectives est $\mathcal{G}$, on a :

$$Rev(G^c_{\mathcal{G}}) \text{ domine } L^i, \forall L^i \in \mathcal{L} \qquad (17)$$

Donc la valeur de Choquet de $Loterie(D_0, \delta)$ est une borne supérieure de celle de la meilleure stratégie complète compatible avec δ.

5 Expérimentations

Les algorithmes proposés ont été implémentés en Java. Ils ont été testés sur un échantillon de 60 arbres de décision, tous de profondeur 6 (chaque stratégie contient cinq décisions). Le nombre d'utilités différentes utilisées par un arbre varie entre 2 et 5 et les valeurs de ces dernières sont comprises entre 0 et 20. Tous les degrés sont interprétés de manière ordinale ($\otimes$ = min). Afin de mieux caractériser le comportement de notre approche nous avons classé les 60 problèmes testés en quatre classes (chacune contenant 15 problèmes) :

- Cl_1 : tous les degrés de possibilité et d'utilité ont été générés aléatoirement.
- Cl_2 : les degrés de possibilité ont été générés aléatoirement et les utilités sont binaires et équiprobables (chaque problème utilise deux utilités).
- Cl_3 : dans chaque loterie le degré de possibilité maximal (1) est affecté à l'utilité maximale.
- Cl_4 : dans chaque loterie le degré de possibilité maximal est affecté à l'utilité minimale.

Pour chaque arbre, nous avons comparé la valeur de Choquet fournie par l'algorithme de programmation dynamique (notée V_{PD}) à celle fournie par l'algorithme de Branch and Bound

[3] Il est évident que $G^c_{Rev(G^c)}(u_1) = 1 = G^c(u_1)$. Notons que $\forall i = 2..n, Rev(G^c)(u_i) \geq Rev(G^c)(u_{i-1})$. D'où $G^c_{Rev(G^c)}(u_i) = 1 - max_{j=1,i-1} Rev(G^c)(u_j) = 1 - Rev(G^c)(u_{i-1})$. Puisque $Rev(G^c)u_{i-1} = 1 - G^c(u_i)$, on a $G^c_{Rev(G^c)}(u_i) = G^c(u_i)$. D'où l'on déduit que $G^c_{Rev(G^c)} = G^c$.

(notée V_{BB}). Le tableau 5 présente pour chaque classe les valeurs de Choquet V_{PD} et V_{BB} moyennes, ainsi que leur ratio $Q = \frac{V_{PD}}{V_{BB}}$ qui estime la qualité de l'approximation fournie par la programmation dynamique. Ces premiers

Classes	V_{PD}	V_{BB}	Q
Cl_1	2.3	6.8	0.33
Cl_2	3.4	3.4	1
Cl_3	3.92	4	0.98
Cl_4	5	5	1

Tableau 1 – Comparaison expérimentale : PD vs. BB

tests confirment que pour les arbres possibilistes standards où les différents degrés sont distribués aléatoirement (classe CL_1), la programmation dynamiques génère des résultats différents de l'algorithme Branch and Bound ; les stratégies fournies par la programmation dynamique ne sont pas proches des stratégies optimales (générées par BB). En revanche, l'approximation est parfaite pour pour les classes Cl_2 et Cl_4 (les arbres avec utilités binaires et ceux où pour chaque loterie le degré de possibilité maximal est affecté à l'utilité minimale). Autrement dit, pour ce type d'arbres, l'algorithme de programmation dynamique peut être utilisé. Pour la troisième classe (les arbres où pour chaque loterie le degré d'utilité maximale a un degré de possibilité égal à 1), V_{PD} et V_{BB} sont très proches : l'algorithme de programmation dynamique génère en général une stratégie optimale. Par ailleurs, le fait que le critère $Q \approx 1$, signifie que dans les rares cas où la stratégie fournie est différente de la stratégie optimale, l'écart entre les deux stratégies n'est pas grand.

6 Conclusion

Dans cet article, nous avons proposé d'utiliser l'intégrale de Choquet basée sur la mesure de nécessité dans des arbres possibilistes avec des utilités numériques compensatoires. Nous avons montré que la programmation dynamique ne conduit pas systématiquement à l'obtention d'une stratégie optimale, la propriété de monotonie sur laquelle cet algorithme s'appuie n'étant pas satisfaite. Nous avons également développé un algorithme de Branch and Bound basé sur une évaluation optimiste de la valeur de Choquet. Les expérimentations nous ont permis de donner une première classification des problèmes pour lesquels la programmation dynamique peut être appliquée. La prochaine étape de ce travail, concerne l'optimisation de l'intégrale de Choquet possibiliste prudente dans les diagrammes d'influence possibilistes hétérogènes qui représentent un modèle graphique compact et facile à utiliser.

Références

[1] D. Dubois, L. Godo, H. Prade, and A. Zapico. Making decision in a qualitative setting : from decision under uncertainty to case-based decision. In *KR'06*, Trento, 1998.

[2] D. Dubois and H. Prade. *Possibility theory, an approach to computerized processing of uncertainty*. Plenum Press, New York, NY, 1988.

[3] D. Dubois and H. Prade. Possibility theory as a basis for qualitative decision theory. In *IJCAI'95*, pages 1924–1930, Montreal, 1995.

[4] L. Garcia et R. Sabbadin. Possibilistic influence diagrams. In *17th ECAI*, pages 372–376, Italie, Août 2006. G. Brewka et col., ed. Riva del Garda.

[5] I. Gilboa. Maxmin expected utility with non unique prior. *Journal of mathematical economics*, 18 :141–153, 1988.

[6] G. Jeantet and O. Spanjaard. Rank-dependent probability weighting in sequential decision problems under uncertainty. In *ICAPS*, pages 148–155, 2008.

[7] H. Fargier N. Ben Amor and W. Guezguez. Necessity-based choquet integrals for sequential decision making under uncertainty. In *Proceedings of IPMU'2010*, pages 222–236, 2010.

[8] J.V. Neumann and O. Morgenstern. *Theory of games and economic behavior*. Princeton University Press, 1948.

[9] H. Fargier R. Sabbadin and J. Lang. Towards qualitative approaches to multi-stage decision making. *IJAR*, 19, 1998.

[10] H. Raiffa. *Decision Analysis : Introductory lectures on choices under uncertainty*. Addison-Wesley, Reading, MA, 1968.

[11] Y. Rébillé. Decision making over necessity measures through the choquet integral criterion. *Fuzzy Sets and Systems*, 157(23) :3025–3039, 12 2006.

[12] R. Sabbadin. A possibilistic model for qualitative sequential decision problems under uncertainty in partially observable environments. In *UAI*, pages 567–574, 1999.

[13] D. Schmeidler. Subjective probability and expected utility without additivity. *Econometrica*, 57 :517–587, 1989.