



HAL
open science

Une approche de choix résolu au sens de Jaffray dans les arbres de décision munis de probabilités imprécises

Gildas Jeantet, Olivier Spanjaard

► To cite this version:

Gildas Jeantet, Olivier Spanjaard. Une approche de choix résolu au sens de Jaffray dans les arbres de décision munis de probabilités imprécises. 11ème Congrès de la Société Française de Recherche Opérationnelle et d'Aide à la Décision (ROADEF 2010), Feb 2010, Toulouse, France. pp.141-154. hal-01291458

HAL Id: hal-01291458

<https://hal.science/hal-01291458>

Submitted on 10 Jul 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Une approche de choix résolu au sens de Jaffray dans les arbres de décision munis de probabilités imprécises

Gildas Jeantet¹, Olivier Spanjaard¹

LIP6-CNRS; UPMC; 104, avenue du Président Kennedy, F-75016 Paris, France
{jeantet,spanjaard}@lip6.fr

Résumé : Dans cet article, nous nous intéressons à l'optimisation du critère d'Hurwicz dans un arbre de décision muni de probabilités imprécises. Nous proposons une procédure de choix résolu au sens de Jaffray [5] afin de déterminer une stratégie dont toutes les sous-stratégies sont à la fois proches de l'optimum (au sens d'Hurwicz) et E -admissible (c'est-à-dire qu'il existe un jeu de probabilités sur le sous-arbre tel que la sous-stratégie maximise l'espérance d'utilité). Cette procédure fait appel à la résolution de programmes linéaires pour évaluer les sous-stratégies selon le critère d'Hurwicz, et pour tester leur E -admissibilité. Des résultats numériques sont fournis qui illustrent le caractère opérationnel de l'approche proposée.

Mots-Clés : décision séquentielle, probabilités imprécises, choix résolu, critère d'Hurwicz.

1 Introduction

La décision dans l'incertain est un des domaines de recherche principaux en théorie de la décision, du fait de ses nombreuses applications (comme par exemple en diagnostic médical, système de contrôle robotique, décision stratégique, jeux...). La décision dans l'incertain traite des problèmes où les conséquences d'une décision dépendent d'événements incertains. Lorsque cette incertitude est probabilisée, on parle alors de décision dans le risque. Dans ce cadre, une décision peut alors être caractérisée par une *loterie* sur les conséquences possibles. Un critère de décision classique pour comparer les loteries (et donc les décisions) est l'espérance d'utilité (EU), introduite par von Neumann et Morgenstern [13]. Dans ce modèle, une *fonction d'utilité* u (propre à chaque décideur) assigne une valeur numérique à chaque conséquence. L'évaluation d'une loterie est alors réalisée via le calcul de son espérance d'utilité (à maximiser). Cependant, lorsque plusieurs experts ont des points de vue divergents ou quand certaines données empiriques manquent, il n'est pas évident d'élucider des probabilités précises pour chacun des événements. Une manière naturelle de prendre en compte explicitement cette difficulté est d'utiliser des intervalles de probabilités en lieu et place des valeurs scalaires de probabilité. On parle alors de *probabilités imprécises*.

Comparer des décisions revient alors à comparer des loteries imprécises, c'est-à-dire des loteries où plusieurs distributions possibles de probabilité sont prises en compte. Un décideur pessimiste prendra la décision qui maximise la pire espérance d'utilité possible. Ceci est connu sous le nom de critère Γ -*maximin*. Inversement, un décideur optimiste prendra la décision qui maximise la meilleure espérance d'utilité possible. C'est le critère Γ -*maximax*. Entre ces deux attitudes extrêmes, Jaffray and Jeleva ont récemment proposé d'utiliser le critère d'Hurwicz qui permet de rendre compte d'attitudes plus

modérées en s'intéressant à une combinaison linéaire des deux critères précédents [2]. Remarquons qu'Hurwicz a initialement introduit ce critère dans un contexte de décision en situation d'ignorance complète (c'est-à-dire lorsqu'aucune information n'est disponible quant aux probabilités d'occurrence des différents événements), mais Jaffray et Jeleva ont décidé de conserver la dénomination de "critère d'Hurwicz" car il s'étend naturellement au cas des probabilités imprécises.

A notre connaissance, les difficultés algorithmiques liées à l'utilisation du critère d'Hurwicz dans les problèmes de décision séquentielle ont été peu étudiées jusqu'à aujourd'hui. Il est en effet fréquent de rencontrer des problèmes de *décision séquentielle* où l'on ne prend pas une unique décision mais où l'on suit plutôt une *stratégie* (c'est-à-dire une séquence de décisions conditionnées par des événements) susceptible de mener à différentes conséquences possibles. Plusieurs formalismes de représentation peuvent être utilisés dans les problèmes de décision séquentielle, tels que les arbres décision hasard [10], les diagrammes d'influence [12] ou encore les processus décisionnels Markoviens [9]. Un arbre de décision est une représentation explicite d'un problème de décision séquentielle, tandis que les diagrammes d'influence ou les processus décisionnels Markoviens sont des représentations compactes qui permettent de traiter des problèmes de décision de plus grande taille. Il est important de remarquer que, dans tous ces formalismes, l'ensemble des stratégies potentielles est combinatoire (c'est-à-dire que sa taille augmente exponentiellement avec la taille de l'instance). Le calcul d'une stratégie optimale pour une représentation donnée et un critère de décision donné est par conséquent un problème algorithmique en soi.

Dans un arbre de décision muni de probabilités scalaires, il est bien connu qu'une stratégie optimale pour EU peut être calculée en temps linéaire par programmation dynamique. Ce n'est plus le cas lorsqu'on utilise des probabilités imprécises et le critère d'Hurwicz. En effet, avec ce critère, une stratégie optimale vue de la racine de l'arbre de décision peut comporter des stratégies sous-optimales dans certains sous-arbres. Dans un précédent article [3], nous avons montré que la détermination d'une stratégie optimale pour le critère d'Hurwicz est un problème NP-difficile dans le cas général, et nous avons proposé un algorithme de résolution opérationnel pour une classe particulière d'arbre de décision. Contournant cette difficulté, dans le cas des critères Γ -maximin et Γ -maximax, Kikuti *et al.* [1] ont présenté des algorithmes qui retournent des stratégies calculées par induction arriérée dans l'arbre de décision, en remontant le temps du futur vers le présent, et en prenant systématiquement la décision qui maximise le critère choisi étant donné les anticipations sur les actions futures. Ce type de stratégie sera suivie par un décideur *sophistiqué*, c'est-à-dire qui est capable d'anticiper sur ses actions futures, et qui adopte de plus un comportement *conséquentialiste*, c'est-à-dire que ses décisions ne dépendent ni du passé, ni d'événements contrefactuels (autrement dit, les événements qui auraient pu se passer). Il a néanmoins été montré que, lorsque les probabilités sont imprécises, l'adoption d'un tel comportement, à la fois sophistiqué et conséquentialiste, rend le décideur vulnérable aux manipulations et plus généralement susceptible de suivre une stratégie "dominée" en un certain sens [4]. Pour cette raison, à la suite de Jaffray [5], nous relâchons ici l'hypothèse de conséquentialisme pour nous intéresser à la détermination d'une stratégie représentant un compromis entre les différents egos d'un décideur sophistiqué. Les *egos* représentent le même décideur dans les diverses situations de choix susceptibles d'apparaître au cours du processus de décision. Chaque ego a ses propres préférences sur les meilleures stratégies à suivre de son point de vue dans son sous-arbre (ces préférences sont induites par l'application du critère de décision choisi). La stratégie de compromis est une stratégie dont chacune des sous-stratégies qui la composent figure parmi l'ensemble des stratégies préférentielles de l'ego correspondant. Contrairement à l'approche développée dans notre précédent travail [3], le choix d'une stratégie ne repose pas uniquement sur le point de vue de l'ego à la racine (qui agit comme une sorte de dictateur), mais tient compte de l'ensemble des points de vue des différents egos. Le décideur suit alors cette stratégie de façon résolue, sans s'autoriser à en dévier au cours du processus de décision. Cette approche a été introduite par McClennen sous le nom de *choix résolu* [8].

Le papier est organisé comme suit. Nous introduisons tout d'abord quelques définitions préliminaires sur les probabilités imprécises, le critère d'Hurwicz et les arbres décision hasard. Après avoir souligné la difficulté de comparer des stratégies au sens du critère d'Hurwicz, nous proposons une procédure de choix résolu. Cette procédure fait en particulier appel à la résolution de programmes linéaires afin de garantir de bonnes propriétés à la stratégie retournée. Enfin, nous présentons des expérimentations numériques illustrant le caractère opérationnel de cette approche.

2 Préliminaires

2.1 Probabilités imprécises et critères de décision

Plusieurs modèles mathématiques de probabilités imprécises ont été proposés dans la littérature [14, 15]. Dans ces différents modèles, un intervalle de probabilité $[P^-(E), P^+(E)]$ est souvent défini pour chaque événement E . A la suite de Jaffray et Jeleva [2], nous supposons qu'il existe une *vraie* probabilité P_0 telle que $P_0(E) \in [P^-(E), P^+(E)]$ pour tout événement E . Pour comparer des loteries imprécises (c'est-à-dire des loteries munies de probabilités imprécises), on doit donc considérer un ensemble \mathcal{P} de distributions de probabilité possibles. Ceci peut être rapproché de ce qui se fait pour comparer des solutions réalisables en optimisation robuste avec données intervalles [6]. La principale différence vient de ce que l'ensemble des distributions de probabilité possibles n'est pas le produit cartésien des intervalles de probabilité des événements élémentaires. Une distribution de probabilité doit en effet respecter les axiomes de Kolmogorov ($P(E) \geq 0$, $P(\Omega) = 1$, $P(E_1 \cup E_2 \cup \dots) = P(E_1) + P(E_2) + \dots$ pour des événements E_i disjoints deux à deux).

Différents critères de décision peuvent être utilisés pour comparer des stratégies en probabilités imprécises. Par exemple, considérons deux loteries f, g faisant intervenir trois événements E_1, E_2, E_3 disjoints deux à deux. Si E_1 (resp. E_2, E_3) se produit, f retourne -50 (resp. 0,100). Parallèlement, si E_1 (resp. E_2, E_3) se produit, g retourne -50 (resp. -30,130). Dans le modèle EU avec des probabilités précises, une loterie est évaluée selon son espérance d'utilité : pour f , celle-ci s'écrit $E(f) = P(E_1)u(-50) + P(E_2)u(0) + P(E_3)u(100)$. Supposons maintenant que les probabilités sont imprécises, avec $P_0(E_1) \in [0.2, 0.4]$, $P_0(E_2) \in [0.4, 0.6]$ et $P_0(E_3) \in [0.2, 0.3]$. L'ensemble \mathcal{P} des distributions de probabilité possible est alors défini par $\mathcal{P} = \{P : P(E_i) \in [P^-(E_i), P^+(E_i)] \forall i, \text{ et } \sum_i P(E_i) = 1\}$. Si le décideur cherche à se prémunir contre la pire espérance d'utilité possible, une loterie f est évaluée par la formule $\underline{E}(f) = \min\{E(f, P) : P \in \mathcal{P}\}$, où $E(f, P)$ est l'espérance d'utilité de la loterie f pour la probabilité P . Ce critère est connu sous le nom de critère Γ -*maximin*. A titre d'illustration, comparons les loteries f et g selon ce critère. La distribution de probabilité P qui minimise $E(f, P)$ est donnée par $P(E_1) = 0.4$, $P(E_2) = 0.4$ et $P(E_3) = 0.2$ (cette distribution peut être calculée aisément par une procédure gloutonne [3]). Par conséquent, pour $u(x) = x$, on a $\underline{E}(f) = 0.4 \times (-50) + 0.2 \times 100 = 0$. De façon similaire, on calcule la distribution P qui minimise $E(g, P) : P(E_1) = 0.2$, $P(E_2) = 0.5$ et $P(E_3) = 0.3$. On en déduit que $\underline{E}(g) = -4$. La loterie f est donc préférée à g pour le critère Γ -*maximin*. Inversement, si le décideur cherche à maximiser la meilleure espérance d'utilité possible, une loterie f est évaluée selon la formule $\bar{E}(f) = \max\{E(f, P) : P \in \mathcal{P}\}$. Il s'agit du critère Γ -*maximax*. A titre d'illustration, considérons à nouveau les loteries f et g définies précédemment. La distribution de probabilité P qui maximise $E(f, P)$ est donnée par $P(E_1) = 0.2$, $P(E_2) = 0.5$ et $P(E_3) = 0.3$. Par conséquent, pour $u(x) = x$, on a $\bar{E}(f) = 20$. De façon similaire, on calcule la distribution P qui maximise $E(g, P) : P(E_1) = 0.4$, $P(E_2) = 0.4$ et $P(E_3) = 0.2$. On en déduit que $\bar{E}(g) = 30$. La loterie g est donc préférée à f pour le critère Γ -*maximax*. Ceci montre que les préférences sont bien évidemment très dépendantes du degré de pessimisme ou d'optimisme du décideur.

Pour cette raison, Jaffray and Jeleva [2] ont étendu le critère d'Hurwicz pour la décision en situation

d'ignorance complète au cas des probabilités imprécises. Avec le critère d'Hurwicz, une loterie f est évaluée par la formule $\alpha \underline{E}(f) + (1 - \alpha) \bar{E}(f)$. En d'autres termes, le décideur se focalise sur la pire et la meilleure espérance d'utilité possible et, selon son degré de pessimisme, accorde d'avantage de poids sur l'une ou l'autre. Ce critère se réduit à Γ -maximin pour $\alpha = 1$, et à Γ -maximax pour $\alpha = 0$. Quand on compare les loteries f et g précédentes selon le critère d'Hurwicz, f est préférée à g pour $\alpha > 5/7$, et g est préférée à f pour $\alpha < 5/7$. Etant donné un ensemble F de loteries définies sur un même espace d'événements élémentaires, notons que le critère d'Hurwicz garantit de sélectionner une loterie f^* non-dominée [2], c'est-à-dire qu'il ne peut exister une loterie $f \in F$ pour laquelle : $E(f, P) \geq E(f^*, P)$ pour toute distribution P et $E(f, P) > E(f^*, P)$ pour au moins une distribution P . Le sous-ensemble des loteries non-dominées dans F est bien évidemment particulièrement pertinent pour n'importe quel décideur rationnel. Un autre sous-ensemble de loteries particulièrement intéressantes aux yeux d'un décideur est celui des loteries E -admissible. Cette notion a été introduite par Levi [7] et discutée par Seidenfeld [11]. Une loterie $f \in F$ est dite E -admissible s'il existe au moins une distribution de probabilité $P \in \mathcal{P}$ pour laquelle elle est maximale dans F au sens de l'espérance d'utilité. Cette dernière propriété garantie des propriétés désirables sur la qualité de la loterie dans l'ensemble F .

2.2 Arbres décision hasard munis de probabilités imprécises

On parle de problème de décision séquentielle lorsqu'on est confronté à un problème où l'on a à prendre une séquence de décision conditionnellement à des événements. Le formalisme des arbres décision hasard fournit une représentation simple et explicite des problèmes de décision séquentielle dans le risque. Il s'agit d'un arbre comportant trois types de nœuds : un ensemble \mathcal{N}_D de nœuds de décision (représentés par des carrés), un ensemble \mathcal{N}_C de nœuds de hasard (représentés par des cercles) et un ensemble \mathcal{N}_U de nœuds d'utilité (les feuilles de l'arbre). A chaque nœud de décision (resp. nœud de hasard) est associé une variable de décision (resp. une variable aléatoire), dont le domaine correspond aux étiquettes des branches issues de ce nœud. Quand les probabilités sont imprécises, la probabilité précise qu'une variable aléatoire donnée prenne une valeur donnée est inconnue : on connaît seulement un intervalle de probabilité dans lequel elle est incluse. Les valeurs indiquées aux feuilles correspondent aux utilités des conséquences. Pour illustration, nous donnons ci-dessous un exemple de problème de décision séquentielle bien connu, et sa représentation sous forme d'un arbre de décision. Remarquons qu'on omet les orientations des arêtes quand on représente un arbre de décision.

Exemple 1 (problème du forage pétrolier [10]) *Une compagnie pétrolière doit décider de forer ou non un site donné. Dans ce but, elle doit d'abord décider si elle sonde ou non la structure géologique du site (décision D_1), ce qui coûte 10000€ et donne une meilleure estimation de la quantité de pétrole qui s'y trouve. Le résultat du sondage peut être vu comme une variable aléatoire X_1 qui peut prendre trois valeurs possibles : aucune si la sonde ne trouve aucune trace de pétrole, faible si la sonde trouve une faible trace de pétrole, ou forte si la sonde trouve une forte trace de pétrole. Ensuite, la compagnie doit décider si elle réalise un forage ou non (décision D_2), ce qui coûte 70000€. Finalement, si la compagnie décide de forer, le résultat du forage peut être vu comme une variable aléatoire X_2 (associée aux nœuds X_2^i pour $i = 1..4$) qui peut prendre trois valeurs possibles : le puit est sec (le revenu est nul), humide (120000€) ou imbibé (270000€). Ce problème peut être représenté par l'arbre de décision sur la gauche de la figure 1. Remarquons que la décision D_2 est dupliquée en plusieurs nœuds (nœuds D_2^1, D_2^2, D_2^3 et D_2^4) car elle peut être prise dans différents contextes (un sondage a été réalisé ou non, le résultat du sondage est encourageant ou non...).*

Choix résolu avec des probabilités imprécises

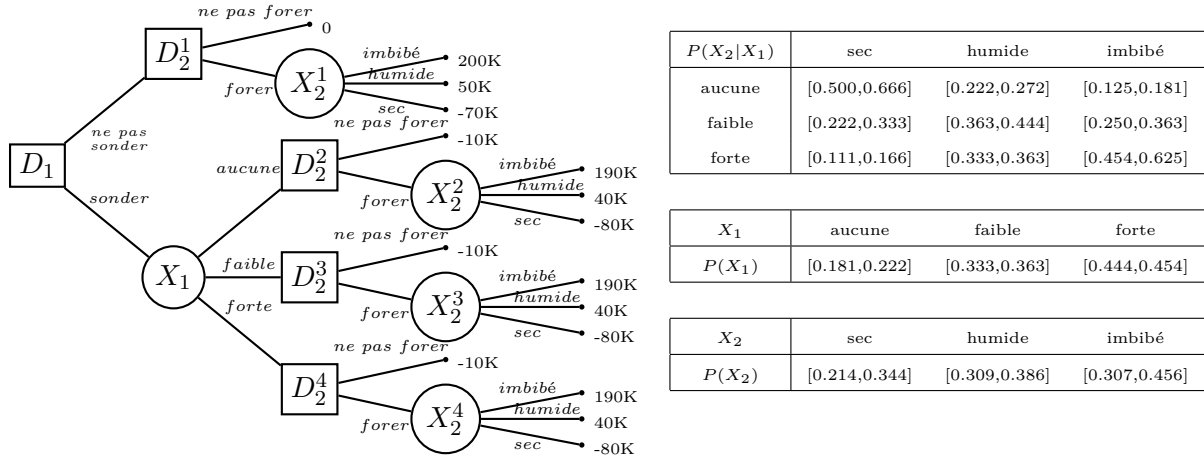


FIG. 1 – Arbre de décision pour le problème du forage pétrolier.

Avant de détailler les difficultés liées à l'utilisation de probabilités imprécises dans un arbre de décision \mathcal{T} , il nous faut introduire au préalable quelques notations qui nous seront utiles par la suite. Soit X_1, \dots, X_n l'ensemble des variables aléatoires figurant dans \mathcal{T} , et $X = \langle X_1, \dots, X_n \rangle$ le vecteur aléatoire correspondant. Nous notons $\mathcal{D}(Y)$ le domaine d'une variable aléatoire (ou vecteur aléatoire) Y . De plus, étant donné un sous-ensemble $I \subseteq \{1, \dots, n\}$ d'indices, nous désignons par X_I le vecteur aléatoire ayant pour composantes les X_i pour $i \in I$. En outre, étant donné un nœud de hasard N , nous désignons par $id(N)$ l'indice i de la variable aléatoire X_i associée au nœud N . De même, étant donné un nœud N , nous désignons par $\pi(N)$ l'ensemble des *indices* des variables aléatoires sur le chemin de la racine à N ($id(N)$ exclu si $N \in \mathcal{N}_C$). L'ensemble complémentaire de ce dernier est noté $\pi^c(N) = \{1, \dots, n\} \setminus \pi(N)$. Enfin, nous désignons par $x(N)$ le vecteur des *valeurs affectées* aux variables aléatoires sur le chemin de la racine à N . Par exemple, sur l'arbre de la figure 1, $id(X_2^2) = 2$ car la variable aléatoire associée à X_2^2 est X_2 , et $\pi(X_2^2) = \{1\}$ car seule la variable aléatoire X_1 figure sur le chemin menant de D_1 à X_2^2 . Par conséquent, on a $X_{\pi(X_2^2)} = \langle X_1 \rangle$, et $x(X_2^2) = \langle aucune \rangle$ car il est nécessaire que X_1 prenne la valeur *aucune* pour arriver en X_2^2 .

Quand les probabilités précises sont connues, chaque branche issue d'un nœud chance N est munie de la probabilité $P(X_{id(N)} = x | X_{\pi(N)} = x(N))$, où $x \in \mathcal{D}(X_{id(N)})$ représente la valeur correspondant à la branche. Remarquons que cette formule n'est valable que sous l'hypothèse que les décisions n'ont pas d'influence sur les distributions de probabilité des variables aléatoires (hypothèse faite dans ce papier). Par exemple, dans l'arbre de décision pour le problème du forage pétrolier, $P(X_2 = imbibé | D_1 = sonder, X_1 = aucune) = P(X_2 = imbibé | X_1 = aucune)$. Quand les probabilités sont imprécises, on suppose qu'une table de probabilités conditionnelles figure en chaque nœud de hasard de l'arbre de décision. Pour le problème du forage pétrolier, les tables de probabilités conditionnelles figurent à côté de l'arbre sur la figure 1.

Une *stratégie* consiste à fixer une valeur à chaque variable de décision conditionnellement à son passé. Nous représenterons une stratégies indifféremment comme un ensemble d'affectation de valeurs à des variables de décision ou comme l'ensemble des arcs de l'arbre correspondant à ces affectations. Par exemple, l'arbre de décision de la figure 1 comporte 10 stratégies réalisables, parmi lesquelles figure, entre autres, la stratégie $s = \{D_1 = sonder, D_2^2 = forer, D_2^3 = forer, D_2^4 = forer\}$ (remarquons que le nœud D_2^1 ne peut pas être atteint quand $D_1 = sonder$), qui peut aussi être notée $\{(D_1, X_1), (D_2^2, X_2^2), (D_2^3, X_2^3), (D_2^4, X_2^4)\}$. Dans notre cadre, une stratégie peut être associée à une loterie composée sur les utilités, où les probabilités des événements impliqués sont imprécises. Par exemple, la stratégie $s = \{D_1 = sonder, D_2^2 = ne pas forer, D_2^3 = forer, D_2^4 = forer\}$ correspond à la loterie composée retournant $-10K$ si $X_1 = aucune$, $190K$ (resp. $40K, -80K$) si $X_1 = faible$ ou $X_1 = forte$ et ensuite $X_2 = imbibé$ (resp. *humide, sec*). Comparer des stratégies se ramène donc à comparer des loteries composées. Etant donné un arbre de décision \mathcal{T} , l'évaluation d'une stratégie

(plus précisément, de la loterie composée correspondante) selon le critère d'Hurwicz dépend de l'ensemble $\mathcal{P}_{\mathcal{T}}$ des distributions de probabilité possibles sur l'arbre de décision \mathcal{T} (c'est-à-dire l'ensemble des affectations de probabilités précises aux tables fournies avec \mathcal{T}). Cette évaluation est un problème combinatoire en soi du fait de la nature combinatoire de $\mathcal{P}_{\mathcal{T}}$.

3 Evaluation d'une stratégie selon le critère d'Hurwicz

Dans cette section, nous nous intéressons à l'évaluation d'une stratégie s selon le critère d'Hurwicz. Il est important de remarquer qu'une stratégie est caractérisée par l'ensemble $\mathcal{U} \subset \mathcal{N}_U$ des noeuds d'utilités qu'elle permet d'atteindre. L'évaluation $V(s)$ d'une stratégie s selon le critère d'Hurwicz peut s'écrire comme suit :

$$V(s) = \alpha \min_{P \in \mathcal{P}_{\mathcal{T}}} \sum_{N \in \mathcal{U}} P(X_{\pi(N)} = x(N)) \times u(N) + (1 - \alpha) \max_{P \in \mathcal{P}_{\mathcal{T}}} \sum_{N \in \mathcal{U}} P(X_{\pi(N)} = x(N)) \times u(N)$$

où $\alpha \in [0, 1]$ et $P(X_{\pi(N)} = x(N))$ représente (conformément à nos notations) le produit des probabilités assignées aux branches le long du chemin de la racine à N . Face à la nature combinatoire du problème, nous allons formuler un programme linéaire pour évaluer la valeur de s selon Γ -maximin (le terme de coefficient α), et un autre programme linéaire pour l'évaluer selon Γ -maximax (le terme de coefficient $1 - \alpha$). Il suffira ensuite de combiner les valeurs obtenues selon le paramètre α pour obtenir l'évaluation selon le critère d'Hurwicz. La principale difficulté consiste alors à caractériser l'ensemble des distributions $P \in \mathcal{P}_{\mathcal{T}}$. En effet, il est important de remarquer que le fait que les probabilités somment à 1 en chaque nœud chance n'est pas suffisant pour garantir la cohérence globale de la distribution de probabilité sur \mathcal{T} . Pour illustration, considérons par exemple la distribution de probabilité partielle suivante sur l'arbre de la figure 1 : $P(X_2 = \text{sec} | X_1 = \text{aucune}) = 0.55$, $P(X_2 = \text{sec} | X_1 = \text{faible}) = 0.33$, $P(X_2 = \text{sec} | X_1 = \text{forte}) = 0.12$, $P(X_1 = \text{aucune}) = 0.20$, $P(X_1 = \text{faible}) = 0.35$, $P(X_1 = \text{forte}) = 0.45$, $P(X_2 = \text{sec}) = 0.22$. Cette distribution partielle peut être complétée de telle sorte que les probabilités somment à 1 en chaque nœud chance, mais est globalement incohérente puisque le théorème des probabilités totales n'est pas vérifié : $P(X_2 = \text{sec} | X_1 = \text{aucune})P(X_1 = \text{aucune}) + P(X_2 = \text{sec} | X_1 = \text{faible})P(X_1 = \text{faible}) + P(X_2 = \text{sec} | X_1 = \text{forte})P(X_1 = \text{forte}) = 0.2795 \neq 0.22 = P(X_2 = \text{sec})$. Afin de caractériser l'ensemble $\mathcal{P}_{\mathcal{T}}$ des probabilités "réalisables", nous allons donc définir un jeu de contraintes linéaires (que nous appellerons *contraintes de cohérence* par la suite) portant sur l'ensemble des variables $P(X = x)$ pour tout x . En effet, cet ensemble de variables suffit à caractériser une distribution de probabilité. Afin de garantir que les axiomes de Kolmogorov sont bien vérifiés, il est nécessaire de satisfaire la contrainte suivante :

$$\sum_{x \in \mathcal{D}(X)} P(X = x) = 1 \quad (1)$$

Par ailleurs, afin de lier les probabilités atomiques $P(X = x)$ aux probabilités $P(X_{\pi(N)} = x(N))$ il est nécessaire d'introduire les contraintes linéaires suivantes pour tout $N \in \mathcal{N}_C \cup \mathcal{N}_U$:

$$P(X_{\pi(N)} = x(N)) = \sum_{y \in \mathcal{D}(X_{\pi^c(N)})} P(X = (x(N), y)) \quad (2)$$

où $(x(N), y)$ désigne l'affectation du vecteur aléatoire X tel que $X_{\pi(N)} = x(N)$ et $X_{\pi^c(N)} = y$. Enfin, pour garantir que les intervalles de probabilités imprécises indiqués dans l'arbre sont bien

respectés, il est nécessaire d'introduire les contraintes suivantes pour tout noeud de hasard N en utilisant les règles de Bayes :

$$P(X_{\pi(N) \cup \{id(N)\}} = (x(N), y)) \leq P(X_{\pi(N)} = x(N))P^+(X_{id(N)} = y | X_{\pi(N)} = x(N)) \quad (3)$$

$$P(X_{\pi(N) \cup \{id(N)\}} = (x(N), y)) \geq P(X_{\pi(N)} = x(N))P^-(X_{id(N)} = y | X_{\pi(N)} = x(N)) \quad (4)$$

Notons que la probabilité $P(X_{\pi(N) \cup \{id(N)\}} = (x(N), y))$ n'est autre que la variable $P(X_{\pi(N')} = (x(N), y))$ dès lors que N est le noeud de hasard précédant immédiatement N' et que l'affectation $X_{id(N)} = y$ figure sur le chemin menant à N' .

Exemple 2 *Considérons l'arbre de décision \mathcal{T} de la figure 1. Nous allons tout d'abord définir l'ensemble des contraintes de cohérence permettant de caractériser $\mathcal{P}_{\mathcal{T}}$. L'ensemble des variables définissant les probabilités atomiques sont : $\{p_{as}, p_{ah}, p_{ai}, p_{fs}, p_{fh}, p_{fi}, p_{Fs}, p_{Fh}, p_{Fi}\}$, où p_{as} correspond à la probabilité $P(X_1 = \underline{aucune}, X_2 = \underline{sec})$ et ainsi de suite (il est à noter que le symbole f est utilisé pour faible, et F pour forte). La contrainte 1 est alors la suivante :*

$$p_{as} + p_{ah} + p_{ai} + p_{fs} + p_{fh} + p_{fi} + p_{Fs} + p_{Fh} + p_{Fi} = 1$$

Ensuite, en chaque noeud de hasard, l'ensemble des contraintes est défini comme suit :

Au noeud de hasard X_2^1 :

- $p_i = p_{ai} + p_{fi} + p_{Fi}$ (contrainte 2)
- $p_h = p_{ah} + p_{fh} + p_{Fh}$ (contrainte 2)
- $p_s = p_{as} + p_{fs} + p_{Fs}$ (contrainte 2)
- $0.307 \leq p_i \leq 0.456$ (contraintes 3 et 4)
- $0.309 \leq p_h \leq 0.386$ (contraintes 3 et 4)
- $0.214 \leq p_s \leq 0.344$ (contraintes 3 et 4)

Au noeud de hasard X_1 :

- $p_a = p_{ai} + p_{ah} + p_{as}$ (contrainte 2)
- $p_f = p_{fi} + p_{fh} + p_{fs}$ (contrainte 2)
- $p_F = p_{Fi} + p_{Fh} + p_{Fs}$ (contrainte 2)
- $0.181 \leq p_a \leq 0.222$ (contraintes 3 et 4)
- $0.333 \leq p_f \leq 0.363$ (contraintes 3 et 4)
- $0.444 \leq p_F \leq 0.454$ (contraintes 3 et 4)

Au noeud de hasard X_2^2 :

- $0.125p_a \leq p_{ai} \leq 0.181p_a$ (contraintes 3 et 4)
- $0.222p_a \leq p_{ah} \leq 0.272p_a$ (contraintes 3 et 4)
- $0.500p_a \leq p_{as} \leq 0.666p_a$ (contraintes 3 et 4)

Au noeud de hasard X_2^3 :

- $0.250p_f \leq p_{fi} \leq 0.363p_f$ (contraintes 3 et 4)
- $0.363p_f \leq p_{fh} \leq 0.444p_f$ (contraintes 3 et 4)
- $0.222p_f \leq p_{fs} \leq 0.333p_f$ (contraintes 3 et 4)

Au noeud de hasard X_2^4 :

- $0.454p_F \leq p_{Fi} \leq 0.625p_F$ (contraintes 3 et 4)
- $0.333p_F \leq p_{Fh} \leq 0.363p_F$ (contraintes 3 et 4)
- $0.111p_F \leq p_{Fs} \leq 0.166p_F$ (contraintes 3 et 4)

Une fois définies les contraintes de cohérences, l'évaluation d'une stratégie selon le critère d'Hurwicz peut être réalisée en définissant les deux programmes linéaires donnant les valeurs selon Γ -maximin et Γ -maximax. Considérons par exemple la stratégie $s = \{D_1 = \text{sonder}, D_2^2 = \text{ne pas forer}, D_2^3 = \text{forer}, D_2^4 = \text{forer}\}$. Les programmes linéaires se formulent comme suit :

$$\Gamma\text{-maximin}(s) = \min \quad -10p_a + 190p_{fi} + 40p_{fh} - 80p_{fs} + 190p_{Fi} + 40p_{Fh} - 80p_{Fs}$$

s.c : contraintes de cohérence

$$\Gamma\text{-maximax}(s) = \max \quad -10p_a + 190p_{fi} + 40p_{fh} - 80p_{fs} + 190p_{Fi} + 40p_{Fh} - 80p_{Fs}$$

s.c : contraintes de cohérence

Enfin la valeur de la stratégie s selon le critère d'Hurwicz est alors :

$$V(s) = \alpha\Gamma\text{-maximin}(s) + (1 - \alpha)\Gamma\text{-maximax}(s)$$

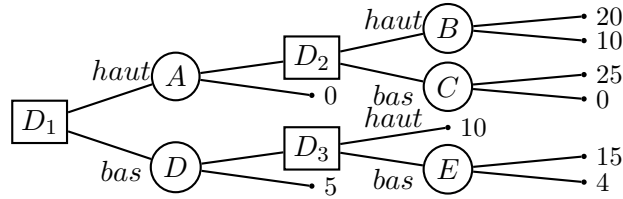


FIG. 2 – Le principe d’optimalité n’est pas vérifié.

D_1	D_2	D_3	$\alpha = 0$	$\alpha = 0.5$	$\alpha = 1$
haut	haut	–	20	10	0
haut	bas	–	25	12.5	0
bas	–	haut	10	7.5	5
bas	–	bas	15	9.5	4

TAB. 1 – Les stratégies et leurs évaluations.

4 Recherche d’une stratégie optimale au sens d’Hurwicz

Lorsque l’on cherche à déterminer une stratégie optimale selon le critère d’Hurwicz dans un arbre de décision, il est important de remarquer que le principe d’optimalité n’est pas vérifié. Par exemple, considérons l’arbre de décision de la figure 2 où les variables aléatoires A, B, C, D, E sont mutuellement indépendantes, et supposons une complète ignorance sur les valeurs des probabilités (c’est-à-dire que tous les intervalles de probabilités sont $[0, 1]$). Posons $\alpha = 0.5$ et réalisons une induction arrière sur l’arbre avec $u(x) = x$. En D_2 , le décideur préfère la décision *haut* à la décision *bas* (le critère d’Hurwicz vaut 15 pour $D_2 = \text{haut}$, contre 12.5 pour $D_2 = \text{bas}$). En D_3 , il préfère également la décision *haut* à *bas* (une utilité certaine de 10, contre 9.5). Enfin, en D_1 , le décideur a le choix entre une première loterie offrant un utilité minimum de 0 et une utilité maximum de 20 si il décide *haut*, et une seconde loterie offrant un minimum de 5 et un maximum de 10 si il décide *bas*. La meilleure décision selon le critère d’Hurwicz est *haut* (10 contre 7.5). La stratégie retournée par programmation dynamique est donc $\{D_1 = \text{haut}, D_2 = \text{haut}\}$ avec une valeur de 10. La table 1 indique les valeurs d’Hurwicz de toutes les stratégies pour différentes valeurs de α . Pour $\alpha = 0.5$, la stratégie $\{D_1 = \text{haut}, D_2 = \text{bas}\}$ est optimale avec une valeur de 12.5. Dans ce cas, on observe donc que la stratégie retournée par programmation dynamique est sous-optimale. Pour cette raison, un décideur utilisant le critère d’Hurwicz doit adopter un comportement de *choix résolu* [8], c’est-à-dire fixer initialement une stratégie puis ne jamais en dévier ensuite. Une manière de faire du choix résolu est de déterminer la stratégie optimale du point de vue de la racine et d’appliquer ensuite la stratégie sans jamais en dévier. Cependant, une critique faite à cette forme de choix de résolu est qu’il n’existe aucune assurance que le décideur ne va pas dévier de la stratégie. Ainsi la stratégie réellement appliquée par le décideur n’a plus aucune garantie de qualité. Face à ce constat, Jaffray [5] a proposé une approche différente de choix résolu. Dans son cadre, on considère chaque noeud de décision comme un *ego* indépendant des autres (différentes “incarnations” d’un même décideur dans plusieurs contextes décisionnels), chaque ego étant maître de la décision qu’il va prendre. Jaffray, dans un tel contexte, propose alors de construire une stratégie qui satisfasse tous les egos présents dans la stratégie. Ainsi, chaque ego, au moment de prendre sa décision, ne sera pas incité à dévier de la stratégie prédéfinie. L’idée est donc d’optimiser le critère de décision parmi l’ensemble des stratégies qui réalisent un compromis acceptable entre les egos. Dans ce travail nous nous intéressons à la détermination d’une telle stratégie.

Les stratégies représentant un compromis acceptable entre les egos sont celles dont toutes les sous-

```

Input : Arbre de décision  $\mathcal{T}$ , réel  $\theta$ , entier  $k$ 
Output : Une stratégie
pour chaque noeud  $N$  de profondeur  $p$  à 0 dans  $\mathcal{T}$  faire
    | si  $N \in \mathcal{N}_D$  alors
    | |  $\mathcal{S}_N \leftarrow \bigcup_{N' \in \text{fils}(N)} \{(N, N')\} \cup s : s \in \mathcal{S}_{N'}\}$ 
    | fin
    | si  $N \in \mathcal{N}_C$  alors
    | |  $\mathcal{S}_N \leftarrow \prod_{N' \in \text{fils}(N)} \mathcal{S}_{N'}$ 
    | fin
    | pour chaque stratégie  $s \in \mathcal{S}_N$  faire
    | |  $V_s \leftarrow \text{evaluer}(s)$ 
    | fin
    |  $V_{max} \leftarrow \max_{s \in \mathcal{S}_N} \{V_s\}$ 
    | pour chaque stratégie  $s \in \mathcal{S}_N$  faire
    | | si ( $s$  n'est pas E-admissible) OU ( $V_s < V_{max} - \theta$ ) alors
    | | |  $\mathcal{S}_N \leftarrow \mathcal{S}_N \setminus \{s\}$ 
    | | fin
    | fin
    | tant que  $|\mathcal{S}_N| > k$  faire
    | |  $\mathcal{S}_N \leftarrow \mathcal{S}_N \setminus \{\arg \min_{s \in \mathcal{S}_N} \{V_s\}\}$ 
    | fin
fin
Retourner  $\{\arg \max_{s \in \mathcal{S}_{racine}} \{V_s\}\}$ 

```

Algorithme 1 : Algorithme pour faire du choix résolu au sens de Jaffray

stratégies sont à la fois E-admissible et proches de l'optimum au sens d'Hurwicz. Nous présentons maintenant une approche procédurale visant précisément à déterminer des stratégies vérifiant ces propriétés. Cette approche est une spécification au cas du critère d'Hurwicz de l'algorithme proposé par Jaffray [5]. Le principe de l'algorithme est de remonter par programmation dynamique, en chaque noeud de décision N , un ensemble de sous-stratégies qui est le résultat d'un compromis entre l'ego associé à N et les egos présents dans le sous-arbre induit par N . Pour ce faire, chaque ego reçoit un ensemble de sous-stratégies estimées acceptables par l'ensemble des egos appartenant à son futur, et met à jour ces sous-stratégies en y préfixant les décisions dont il dispose. Une première condition nécessaire à l'adhésion d'un ego à une stratégie de compromis est l'E-admissibilité de la sous-stratégie induite. En effet, il est difficilement concevable qu'un ego puisse appliquer une stratégie qui n'est jamais optimale quelle que soit la vraie distribution de probabilité. C'est pourquoi on rejette toutes les sous-stratégies qui ne sont pas E-admissibles. Chaque sous-stratégie de l'ensemble est ensuite évaluée selon le critère d'Hurwicz. Soient V_{max} la meilleure évaluation de l'ensemble et θ la tolérance de perte d'utilité pour chaque ego (θ représente le degré de compromis acceptable par les egos afin de garantir qu'ils ne dévieront pas de la stratégie prédéfinie). Toutes les sous-stratégies dont l'évaluation est strictement inférieure à $V_{max} - \theta$ sont rejetées. Enfin, parmi les sous-stratégies E-admissibles dont la valeur est supérieure ou égale à $V_{max} - \theta$, on sélectionne les k meilleures afin de se préserver d'une explosion combinatoire de la taille des ensembles de sous-stratégies remontées en chaque noeud. Une fois l'induction arrière terminée, on sélectionne la stratégie de meilleure évaluation dans l'ensemble remonté à la racine de l'arbre et on la retourne. Notons que, dans certains cas, cet ensemble peut être vide et l'algorithme ne retourne alors aucune solution. Dans ces cas, il est nécessaire de réviser les valeurs des paramètres θ et/ou k si les ressources physiques de calcul le permettent. Ainsi, la stratégie

retournée par la procédure comporte uniquement des sous-stratégies E-admissibles et proches de l'optimum au sens du critère d'Hurwicz. Elle est bien sûr elle-même globalement E-admissible. La procédure est formalisée dans l'algorithme 1, où p désigne la profondeur de l'arbre, \mathcal{S}_N (resp. $fil_s(N)$) l'ensemble des sous-stratégies remontées en un nœud N (resp. l'ensemble des fils du nœud N), et **evaluer**(s) retourne la valeur de s selon le critère d'Hurwicz en résolvant les programmes linéaires définis comme indiqués en section 3.

L'algorithme 1 parcourt chaque nœud N en partant des feuilles de l'arbre jusqu'à la racine. En chacun de ces nœuds, un ensemble \mathcal{S}_N de stratégies est construit à partir des ensemble $\mathcal{S}_{N'}$ de ses fils. Ensuite, chaque stratégie est évaluée (avec la fonction **evaluer**(s)). Enfin, on ne conserve dans l'ensemble \mathcal{S}_N que les k meilleures stratégies qui sont E-admissible, et dont l'évaluation est supérieure à $V_{max} - \theta$. Enfin, une fois l'ensemble de stratégies construit en la racine, on retourne celle d'évaluation maximale. Pour conclure cette partie algorithmique, il nous reste à expliquer comment déterminer si une stratégie est E-admissible ou pas.

Rappelons qu'une stratégie s est dite *E-admissible* s'il existe au moins une distribution de probabilité $P \in \mathcal{P}_{\mathcal{T}}$ pour laquelle la stratégie s est maximale au sens de l'espérance d'utilité. Nous montrons maintenant comment réduire le test d'E-admissibilité d'une stratégie à un problème de satisfaction de contraintes linéaires. Plus précisément, la stratégie s sera E-admissible si et seulement si le polyèdre défini par l'ensemble des contraintes est non-vide. L'ensemble des contraintes comporte bien évidemment les contraintes de cohérence (sur les probabilités) définies en section 3. Un autre jeu de contraintes caractérise l'ensemble des stratégies réalisables. Pour ce faire, on introduit une variable réelle y_N pour chaque nœud de décision N (cette variable est destinée à représenter l'espérance d'utilité maximale en N) et on définit les contraintes suivantes :

$$y_N = P\left(X_{\pi(N)} = x(N)\right) \times u(N) \quad \forall N \in \mathcal{N}_U \quad (5)$$

$$y_N = \sum_{N' \in fil_s(N)} y_{N'} \quad \forall N \in \mathcal{N}_C \quad (6)$$

Enfin, un dernier jeu de contraintes impose que la stratégie s soit maximale au sens de l'espérance d'utilité. Pour chaque arc (N, N') issu d'un nœud de décision N :

- si (N, N') appartient à la stratégie s , on ajoute la contrainte

$$y_N \leq y_{N'} \quad (7)$$

- si (N, N') n'appartient pas à la stratégie s , on ajoute la contrainte

$$y_N \geq y_{N'} \quad (8)$$

Ce dernier jeu de contraintes impose que y_N vaut au plus l'espérance d'utilité de la sous-stratégie induite par s dans le sous-arbre de racine N , et au moins l'espérance d'utilité maximum dans le sous-arbre de racine N . On en conclut que si une telle espérance existe en chaque nœud de décision, alors il existe une distribution de probabilité $P \in \mathcal{P}_{\mathcal{T}}$ (caractérisée par les variables $P(X = x)$) telle que l'espérance d'utilité de s soit de valeur maximale. Pour illustration, nous indiquons ci-dessous les contraintes obtenues pour une stratégie particulière dans l'exemple du forage pétrolier.

Exemple 3 *Considérons l'arbre de décision hasard de la figure 1 et la stratégie $s = \{D_1 = \text{sonder}, D_2^2 = \text{ne pas forer}, D_2^3 = \text{forer}, D_2^4 = \text{forer}\}$. Les contraintes de cohérence pour cet arbre sont données dans l'exemple 2. Par souci de brièveté, nous n'indiquons pas ici les contraintes et les variables associées aux nœuds d'utilité. Les contraintes en chacun des autres nœuds sont les suivantes :*

- Au noeud X_2^1 : $y_{X_2^1} = 200p_i + 50p_h - 70p_s$
- Au noeud D_2^1 : $y_{D_2^1} \geq 0$ et $y_{D_2^1} \geq y_{X_2^1}$
- Au noeud X_2^2 : $y_{X_2^2} = 200p_{ai} + 50p_{ah} - 70p_{as}$
- Au noeud D_2^2 : $y_{D_2^2} \leq -10p_a$ et $y_{D_2^2} \geq y_{X_2^2}$
- Au noeud X_2^3 : $y_{X_2^3} = 200p_{fi} + 50p_{fh} - 70p_{fs}$
- Au noeud D_2^3 : $y_{D_2^3} \geq -10p_f$ et $y_{D_2^3} \leq y_{X_2^3}$
- Au noeud X_2^4 : $y_{X_2^4} = 200p_{Fi} + 50p_{Fh} - 70p_{Fs}$
- Au noeud D_2^4 : $y_{D_2^4} \geq -10p_F$ et $y_{D_2^4} \leq y_{X_2^4}$
- Au noeud X_1 : $y_{X_1} = y_{D_2^2} + y_{D_2^3} + y_{D_2^4}$
- Au noeud D_1 : $y_{D_1} \geq y_{D_2^1}$ et $y_{D_1} \leq y_{X_1}$

5 Expérimentations numériques

Afin d'illustrer les performances de l'algorithme proposé ici, nous avons mené des expérimentations numériques sur des arbres décision hasard générés aléatoirement. Les arbres générés ont pour racine un noeud de décision et alternent noeud de décision et noeud de hasard sur toutes les branches pour s'achever par un noeud de hasard puis un noeud d'utilité. Notons également que les arbres générés sont des arbres binaires complets, c'est-à-dire que chaque noeud a exactement 2 fils (et 0 si le noeud est de profondeur max). Les utilités sont des réels tirés aléatoirement entre 0 et 500. Ensuite, à chaque noeud de hasard est associée une variable aléatoire. En ce qui concerne la génération des probabilités, nous générons d'abord l'ensemble des probabilité atomiques exactes de manière aléatoire afin de s'assurer qu'au moins une distribution de probabilité existe. A partir de cette distribution, en suivant les règles de Bayes, nous calculons les probabilités précises conditionnelles en chaque noeud de hasard. Enfin nous générons aléatoirement un intervalle autour de cette probabilité précise, qui constitue notre probabilité imprécise. Les expérimentations numériques ont été lancées sur un Pentium IV avec un processeur à 2,13GHz. L'algorithme principal a été implémenté en C++ et les programmes linéaires ont été résolus à l'aide de CPLEX 11.

		$p = 4$	$p = 6$	$p = 8$	$p = 10$	$p = 12$	$p = 14$
$k = 2$	$\theta = 10$	< 0.01	0.04	0.39	6.22	111.73	2518.95
$k = 2$	$\theta = 50$	< 0.01	0.04	0.38	6.24	112.78	2487.56
$k = 2$	$\theta = 100$	< 0.01	0.04	0.39	6.18	112.34	2660.13
$k = 2$	$\theta = 300$	< 0.01	0.04	0.37	6.23	110.15	2532.40
$k = 5$	$\theta = 10$	< 0.01	0.04	0.38	6.34	114.10	2711.24
$k = 5$	$\theta = 50$	< 0.01	0.04	0.40	6.42	115.21	2689.83
$k = 5$	$\theta = 100$	< 0.01	0.04	0.38	6.39	115.75	2802.88
$k = 5$	$\theta = 300$	< 0.01	0.04	0.38	6.41	117.60	2765.12
$k = 10$	$\theta = 10$	< 0.01	0.04	0.39	6.78	121.35	3398.91
$k = 10$	$\theta = 50$	< 0.01	0.04	0.37	6.82	126.03	2991.33
$k = 10$	$\theta = 100$	< 0.01	0.04	0.38	6.91	124.53	3132.48
$k = 10$	$\theta = 300$	< 0.01	0.04	0.38	6.87	125.84	3255.04

TAB. 2 – Moyenne du temps d'exécution en seconde en fonction de la profondeur p de l'arbre et des paramètres k et θ .

Pour la première série d'expérimentations numériques, nous avons traité des arbres décision hasard où chaque niveau de noeud de hasard est associé à une même variable aléatoire. Remarquons que dans ce cadre, le nombre de probabilités atomiques ($\in O(2^v)$ si v est le nombre de variables aléatoires) est alors linéaire en la taille de l'instance puisque v est alors une valeur logarithmique du nombre n de noeuds dans l'arbre. Ceci implique une complexité polynomiale de l'algorithme en fonction de k et n . Pour cette série d'expérimentations nous avons fait varier k , θ et p (la profondeur de l'arbre). Nous nous sommes intéressés aux temps d'exécution de l'algorithme en fonction de ces paramètres. Pour chaque jeu de paramètres où $p \leq 12$, 100 arbres ont été générés aléatoirement et nous avons indiqué

	$p = 5$	$p = 7$	$p = 9$	$p = 11$	$p = 13$
$v = 2$	< 0.01	–	–	–	–
$v = 3$	< 0.01	0.04	–	–	–
$v = 4$	< 0.01	0.04	0.40	–	–
$v = 5$	< 0.01	0.05	0.66	6.42	–
$v = 6$	< 0.01	0.05	0.85	7.65	115.21
$v = 7$	< 0.01	0.08	1.18	10.87	135.28
$v = 8$	< 0.01	0.13	1.41	13.71	214.21
$v = 9$	0.02	0.17	1.86	21.96	287.68
$v = 10$	0.03	0.25	3.44	35.61	485.16
$v = 11$	–	0.56	6.25	62.95	698.81
$v = 12$	–	0.94	11.40	94.35	X
$v = 13$	–	1.92	22.10	165.61	X
$v = 14$	–	4.03	42.03	342.91	X
$v = 15$	–	9.58	80.01	680.95	X
$v = 16$	–	20.98	162.46	X	X
$v = 17$	–	49.11	347.77	X	X
$v = 18$	–	98.06	786.34	X	X

TAB. 3 – Moyenne du temps d’exécution en seconde en fonction de la profondeur p de l’arbre et du nombre de variable aléatoire v .

sur la table 2 la moyenne du temps d’exécution en secondes. Pour $p = 14$, seulement 5 instances ont été générées aléatoirement pour les différents paramètres. Les temps d’exécution dépendent évidemment de k et θ puisqu’ils conditionnent le nombre de stratégies retenues en chaque noeud. Avec ce type d’instances, nous sommes capables de résoudre rapidement (en moins d’une heure) des instances jusqu’à la profondeur 14 (ce qui correspond à 65535 noeuds dans l’arbre).

Dans la seconde série d’expérimentations numériques, nous nous sommes intéressés aux arbres décision hasard où le nombre de variables aléatoires distinctes dans l’arbre de décision hasard est indépendant de la taille de l’instance. Pour cette série d’expérimentations nous avons fait varier p et v . Nous nous sommes intéressés à l’influence du nombre de variables sur le temps d’exécution. Nous avons donc fixé le paramètre k à 5 et le paramètre θ à 50. Pour chaque jeu de paramètres, nous avons généré 100 instances aléatoirement en attribuant de manière aléatoire les variables aléatoires aux noeuds de hasard. Les temps indiqués sur la table 3 représentent la moyenne du temps d’exécution en secondes pour chaque jeu de paramètres (le symbole “–” indique que l’instance est impossible et le symbole “ X ” correspond aux jeux de paramètres impliquant un temps de résolution supérieur à 1000 secondes). Nous constatons qu’à partir d’un certain nombre de variables aléatoires, l’ajout d’une variable aléatoire supplémentaire à l’instance double le temps d’exécution moyen de l’algorithme.

6 Conclusion

Dans cet article, nous avons présenté une procédure de choix résolu au sens de Jaffray pour calculer une stratégie dont toutes les sous-stratégies sont à la fois E-admissibles et proches de l’optimum au sens du critère d’Hurwicz. La principale difficulté vient de la nature combinatoire de l’ensemble des distributions de probabilité réalisables sur l’arbre de décision. Face à cette difficulté, nous avons

formulé des programmes linéaires qui permettent de tenir compte explicitement des contraintes sur les probabilités, que ce soit pour évaluer une stratégie au sens du critère d'Hurwicz ou pour tester l'E-admissibilité d'une stratégie. Les résultats sont assez encourageants, en effet, il est possible de traiter des instances d'arbres complets (arbres dont la combinatoire est la plus élevée) de plus de 65000 noeuds. Pour des travaux futurs, il serait intéressant de viser à garantir une propriété plus forte que l'E-admissibilité des sous-stratégies de la stratégie retournée. En effet, une sous-stratégie peut être E-admissible mais néanmoins dominée (au sens de la dominance définie en section 2.1). Par exemple, considérons un ensemble $\{f, g\}$ de loteries sur les événements E_1 et E_2 . Supposons que $P_0(E_1) \in [0, 0.1]$ et $P_0(E_2) \in [0.9, 1]$. La loterie f retourne 100 si E_1 ou E_2 se produit. La loterie g quant à elle retourne 0 si E_1 se produit et 100 si E_2 se produit. Clairement, la loterie f domine g , bien que g soit E-admissible (pour $P(E_2) = 1$). C'est pourquoi nous pensons qu'il serait intéressant de proposer une méthode efficace pour tester qu'une stratégie est non-dominée, et de l'intégrer dans une procédure de choix résolu comme celle présentée ici (en remplacement du test d'E-admissibilité).

A la mémoire de Jean-Yves Jaffray, collègue et ami.

Références

- [1] F. G. Cozman et C. P. de Campos D. Kikuti. Partially ordered preferences in decision trees : computing strategies with imprecision in probabilities. In *IJCAI Workshop on Advances in Preference Handling*, 2005.
- [2] J.-Y. Jaffray et M. Jeleva. Information processing under imprecise risk with the Hurwicz criterion. In *5th International Symposium on Imprecise Probability : Theories and Applications*, pages 233–242, 2007.
- [3] G. Jeantet et O. Spanjaard. Optimizing the Hurwicz criterion in decision trees with imprecise probabilities. In *International Conference on Algorithmic Decision Theory*, pages 340–352. Springer-Verlag, 2009.
- [4] P. Hammond. Consequentialist foundations for expected utility. *Theory and Decision*, 25 :25–78, 1988.
- [5] J.-Y. Jaffray. Rational decision making with imprecise probabilities. In *1st International Symposium on Imprecise Probability : Theories and Applications*, pages 183–188, 1999.
- [6] A. Kasperski. *Discrete Optimization with Interval Data : Minmax Regret and Fuzzy Approach*. Studies in Fuzziness and Soft Computing. Springer, 2008.
- [7] I. Levi. On indeterminate probabilities. *Journal of Philosophy*, 71(13) :391–418, 1974.
- [8] E.F. McClennen. *Rationality and Dynamic choice : Foundational Explorations*. Cambridge University Press, 1990.
- [9] M.L. Puterman. *Markov Decision Processes - Discrete Stochastic Dynamic Programming*. Wiley & Sons, 1994.
- [10] H. Raiffa. *Decision Analysis : Introductory Lectures on Choices under Uncertainty*. Addison-Wesley, 1968.
- [11] T. Seidenfeld. A contrast between two decision rules for use with (convex) sets of probabilities : G-maximin versus e-admissibility. *Synthèse*, 140 :69–88, 2004.
- [12] R. Shachter. Evaluating influence diagrams. *Operations Research*, pages 34 :871–882, 1986.
- [13] J. von Neuman et O. Morgenstern. *Theory of games and economic behaviour*. Princeton University Press, 1947.

- [14] P. Walley. *Statistical reasoning with imprecise probabilities*, volume 91 of *Monographs on statistics and applied probability*. Chapman and Hall, 1991.
- [15] K. Weichselberger. The theory of interval-probability as a unifying concept for uncertainty. In *1st International Symposium on Imprecise Probability : Theories and Applications*, pages 387–396, 1999.