

ajout de $0.9u(0)$ de chaque côté de l'inéquation, à $0.9u(0) + 0.1u(3000) > 0.91u(0) + 0.09u(4000)$. Cette dernière inégalité traduit la préférence de L_2 sur L'_2 . Ainsi, quelle que soit la fonction d'utilité utilisée, la préférence pour L_1 sur L'_1 est équivalente à la préférence pour L_2 sur L'_2 dans le cadre du modèle EU.

Allais qualifie ce comportement, loin d'être paradoxal, comme un comportement raisonnable de "préférence pour la sécurité dans le voisinage de la certitude" [2]. En d'autres termes, "un tiens vaut mieux que deux tu l'auras". Ce phénomène est aujourd'hui bien connu sous le nom d'*effet de certitude*. Dans l'exemple, la probabilité de gagner 3000 € dans L_1 (resp. 4000 € dans L'_1) est simplement multipliée par 0.1 dans L_2 (resp. L'_2). Le renversement de préférence peut être expliqué comme suit : quand la probabilité de gagner devient basse, la sensibilité aux probabilités décroît. Pour tenir compte de l'effet de certitude dans l'élaboration d'un critère de choix, la traitement des probabilités ne doit donc pas être linéaire. Ceci a conduit à l'émergence de modèles non-linéaires plus sophistiqués qui généralisent EU. A ce titre, nous nous intéressons ici plus particulièrement au modèle de l'*utilité espérée dépendant du rang* (RDU pour *rank dependent utility*) [18] qui généralise le modèle EU. Dans ce modèle, une fonction non-linéaire φ de déformation des probabilités est introduite, ce qui permet de rendre compte de comportement décisionnel tel que celui observé dans le paradoxe d'Allais. De plus, la fonction φ de déformation des probabilités permet aussi de modéliser plus finement l'attitude du décideur vis-à-vis du risque. En effet, contrairement au modèle EU, le modèle RDU permet de distinguer entre l'aversion au risque (i.e., une conséquence certaine est toujours préférée à une loterie dont l'espérance d'utilité est égale à l'utilité de la conséquence certaine) et l'aversion à un accroissement de risque (i.e., si deux loteries ont la même espérance d'utilité, l'agent préfère toujours celle dont l'étalement des conséquences possibles est le plus faible). Ce soucis de modéliser finement l'attitude de l'agent vis-à-vis du risque a été mis en avant dans plusieurs travaux récents en IA [16, 13].

La difficulté de mise en œuvre opérationnelle du modèle RDU a freiné son utilisation jusqu'à aujourd'hui. En effet, dans la pratique, il est fréquent de rencontrer des problèmes de décision *séquentielle* dans le risque, c'est-à-dire des problèmes où l'on ne suit pas une unique décision, mais une *stratégie* (i.e. une séquence

de décisions conditionnellement à des événements incertains) conduisant à une conséquence finale incertaine. C'est le cas en particulier dans les problèmes de planification dans l'incertain étudiés en IA, tels que la navigation d'un robot mobile, le diagnostic médical, l'organisation des secours ou encore la planification des actions d'un joueur artificiel. Dans de tels problèmes, on cherche à établir un plan optimal, ce qui nous amène à comparer des stratégies. Pour ce faire, on associe à chaque stratégie la loterie induite sur les conséquences potentielles, puis on compare ces loteries à l'aide d'un critère de décision (EU usuellement). Ces problèmes peuvent se représenter à l'aide de plusieurs outils formels, parmi lesquels on peut mentionner entre autres les arbres décision-hasard [19], les diagrammes d'influence [7] et les processus décisionnels markoviens [4]. Il est important de noter que dans toutes ces représentations l'ensemble des stratégies potentielles est combinatoire. La recherche d'une stratégie optimale pour une représentation et un critère de décision donnés est alors un problème algorithmique en soi. La non-linéarité du critère RDU induit de nouvelles difficultés algorithmiques, qui ne se posaient pas avec le critère EU. Ceci fait précisément l'objet de notre travail.

Un premier travail a déjà été mené sur l'optimisation de RDU dans les arbres décision hasard [10] (le problème a été prouvé NP-difficile, et un algorithme d'énumération implicite a été proposé). Toutefois, cette approche se heurte à l'inconvénient principal des arbres décision hasard : leur taille peut rapidement devenir prohibitive lorsque le nombre d'étapes de décision croît. Pour cette raison, nous nous intéressons ici à l'optimisation de RDU dans le cadre des *diagrammes d'influence* [7]. En effet, les diagrammes d'influence sont des modèles graphiques qui permettent une représentation particulièrement compacte des problèmes de décision séquentielle dans l'incertain, en exploitant les dépendances probabilistes du problème pour factoriser certaines données. La détermination d'une stratégie EU-optimale dans un diagramme d'influence est déjà un problème NP-difficile. Néanmoins des algorithmes performants utilisant des techniques importées des réseaux bayésiens existent pour le résoudre [11]. Ces algorithmes ne s'étendent pas pour l'optimisation du critère RDU. De plus, une difficulté supplémentaire survient lorsque l'on utilise ce dernier critère : contrairement à une stratégie EU-optimale, une stratégie RDU-optimale n'est pas nécessairement *conséquentialiste*. En

toute rigueur, un *comportement* est dit conséquentialiste lorsque les décisions de l'agent ne dépendent ni du passé ni de scénarios contre-factuels (scénarios qui auraient pu se produire, mais cela n'a pas été le cas) [8]. Par abus de langage, nous qualifions ici de conséquentialiste une *stratégie* où chaque décision prise ne dépend que des variables influençant les paramètres futurs (probabilités et conséquences). Dans les diagrammes d'influence, cette propriété est cruciale car la description de telles stratégies reste alors linéaire dans la taille du diagramme, ce qui n'est plus le cas pour les stratégies non-conséquentialistes. La description d'une stratégie RDU-optimale peut ainsi nécessiter un espace mémoire exponentiel dans la taille de l'instance, ce qui rend la tâche d'optimisation singulièrement ardue.

Le papier est organisé comme suit : dans la section 2, nous rappelons les bases du modèle RDU ; dans la section 3 nous rappelons le formalisme des diagrammes d'influence et nous mettons en avant les difficultés soulevées par l'utilisation de RDU ; dans la section 5 nous présentons deux approches visant à optimiser RDU dans un diagramme d'influence ; enfin dans la section 6 nous présentons les résultats numériques que nous avons obtenus.

2 Utilité espérée dépendant du rang

Dans le modèle RDU, le poids attribué à une conséquence ne dépend pas uniquement de la probabilité d'obtenir cette conséquence, mais également de l'attractivité du rang de cette conséquence par rapport aux autres conséquences possibles. Ce modèle repose sur deux paramètres : une fonction d'utilité qui est déjà présente dans le modèle EU, et une fonction φ de déformation des probabilités. Il s'agit d'une fonction strictement croissante sur $[0, 1]$ telle que $\varphi(0) = 0$ et $\varphi(1) = 1$. Cette déformation des probabilités porte, non sur des probabilités simples, mais sur des cumuls de probabilités. Pour rappel, étant donnée une loterie $L = (p_1, u_1; \dots; p_k, u_k)$ (qui rapporte une utilité u_i avec une probabilité p_i), on appelle *fonction décumulative* de L la fonction $G_L : S \mapsto [0, 1]$ qui associe à chaque utilité u_i la probabilité d'avoir au moins cette utilité. Plus formellement, $G_L(x) = \sum_{i:u_i \geq x} p_i$. La valeur selon RDU d'une loterie L est alors définie de la manière suivante :

$$RDU(L) = u_{(1)} + \sum_{i=2}^k [u_{(i)} - u_{(i-1)}] \varphi(G_L(u_{(i)}))$$

où $(.)$ correspond à une permutation de $\{1, \dots, k\}$ telle que $u_{(1)} \leq \dots \leq u_{(k)}$. Ce critère peut être interprété comme suit : on est sûr d'obtenir au moins une utilité de $u_{(1)}$, puis on est susceptible d'obtenir un supplément d'utilité de $u_{(2)} - u_{(1)}$ avec une masse de probabilité $\varphi(G_L(u_{(2)}))$, puis un supplément d'utilité de $u_{(3)} - u_{(2)}$ avec une masse de probabilité $\varphi(G_L(u_{(3)}))$, et ainsi de suite... Ce modèle est en fait une généralisation du modèle EU. En effet, si la fonction de déformation des probabilités est linéaire, alors on se retrouve dans le cadre du modèle EU. Le modèle RDU permet de résoudre le paradoxe d'Allais (voir exemple 2) et bien d'autres situations où le modèle EU était impuissant.

Exemple 2 Revenons sur le paradoxe d'Allais. On définit pour l'agent une utilité linéaire $u(x) = x$ et on utilise la fonction φ suivante : $\varphi(0.09) = \varphi(0.1) = 0.2, \varphi(0.9) = 0.7$. Les préférences retournées par le modèle RDU sont alors compatibles avec le paradoxe d'Allais :

$$\begin{aligned} RDU(L_1) &= u(3000) \\ &= 3000 \\ RDU(L'_1) &= u(0) + \varphi(0.9)(u(4000) - u(0)) \\ &= 2800 \end{aligned}$$

Ainsi L_1 est préférée à L'_1 . De même on a :

$$\begin{aligned} RDU(L_2) &= u(0) + \varphi(0.1)(u(3000) - u(0)) \\ &= 600 \\ RDU(L'_2) &= u(0) + \varphi(0.09)(u(4000) - u(0)) \\ &= 800 \end{aligned}$$

Et donc L'_2 est préférée à L_2 .

L'intérêt de déformer des cumuls de probabilités, et non directement les probabilités elles-mêmes (comme c'est par exemple le cas dans le modèle de Handa [6]), est d'obtenir un critère de choix compatible avec la dominance stochastique. On dit qu'une loterie $L = (p_1, u_1; \dots; p_k, u_k)$ domine stochastiquement une loterie $L' = (p'_1, u'_1; \dots; p'_k, u'_k)$ si $\forall x \in \mathbb{R}, G_L(x) \geq G_{L'}(x)$, autrement dit, pour tout $x \in \mathbb{R}$, la probabilité d'obtenir une utilité d'au moins x avec la loterie L est au moins aussi grande qu'avec la loterie L' . La compatibilité avec la dominance stochastique signifie que $RDU(L) \geq RDU(L')$ dès lors que L domine stochastiquement L' [17]. Cette propriété est bien entendu souhaitable pour décrire un comportement rationnel, et elle est bien vérifiée par le modèle RDU (contrairement au modèle de Handa).

3 Diagramme d'influence

Le formalisme des *arbres décision hasard* fournit une représentation simple et explicite d'un problème de décisions séquentielles dans le risque. Il s'agit d'une arborescence comportant trois types de nœuds : les *nœuds de décision* (représentés par des carrés), les *nœuds de hasard* (représentés par des cercles), et les *nœuds terminaux* (les feuilles de l'arborescence). Les branches issues d'un nœud de décision correspondent à différentes décisions possibles, tandis que celles issues d'un nœud de hasard correspondent aux différents événements possibles, dont les probabilités sont indiquées sur les branches. Enfin, les valeurs figurant au niveau des feuilles de l'arborescence correspondent aux utilités des différentes conséquences. Un exemple d'arbre décision hasard est fourni sur la figure 2.

Un *diagramme d'influence* [7] est une représentation graphique alternative d'un problème de décisions séquentielles. A la différence d'un arbre décision hasard, l'accent est mis sur la décomposabilité de la structure de probabilité sous-jacente au problème. En effet, en tirant parti des indépendances entre les variables aléatoires et les variables d'utilité impliquées, on obtient une représentation beaucoup plus compacte que celle obtenue en explicitant tous les cas de figure possibles, comme on le ferait dans un arbre décision hasard. Un diagramme d'influence incluant des variables aléatoires A_1, \dots, A_n et des variables de décision D_1, \dots, D_k est un graphe orienté sans circuit $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ tel que :

- l'ensemble \mathcal{N} se partitionne en trois sous-ensembles : un ensemble $\mathcal{N}_D = D_1, \dots, D_n$ de variables de décision (représentées par des carrés), un ensemble $\mathcal{N}_A = A_1, \dots, A_p$ de variables aléatoires (représentées par des cercles), et un ensemble $\mathcal{N}_U = U_1, \dots, U_m$ de nœuds d'utilité (représentés par des losanges) ;
- l'ensemble \mathcal{E} des arcs se partitionne en deux sous-ensembles : un ensemble d'arcs fonctionnels allant d'une variable de décision ou d'une variable aléatoire vers une variable aléatoire ou un nœud d'utilité (arcs représentant des dépendances), un ensemble d'arcs informationnels allant d'une variable de décision ou d'une variable aléatoire vers une variable de décision (arcs représentant les variables observées avant de prendre une décision) ;
- les nœuds représentant les variables aléatoires

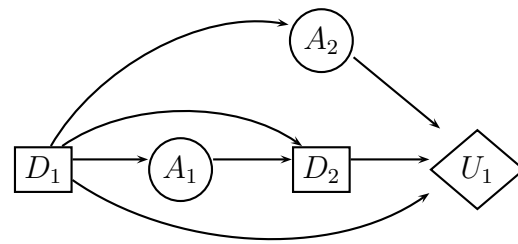


FIG. 1 – Diagramme d'influence.

sont munis d'une table de probabilité conditionnelle, qui indique la probabilité de chaque événement conditionnellement aux nœuds parents ;

- les nœuds d'utilité sont munis d'une table indiquant l'utilité conditionnellement aux nœuds parents.

Le graphe doit de plus satisfaire la condition structurelle suivante : il existe un chemin (comportant des arcs fonctionnels et informationnels) connectant tous les nœuds représentant les variables de décision.

Un diagramme d'influence est représenté sur la figure 1, où l'on a omis volontairement les tables de probabilité conditionnelle et d'utilité par soucis de brièveté. L'ordre des décisions et des observations est $D_1 - A_1 - D_2 - A_2$. Nous adoptons ici la convention que les variables de décision sont disposées sur le graphe de la gauche vers la droite dans l'ordre chronologique. En développant le diagramme, on retrouve l'arbre décision hasard de la figure 2 (remarquons que les probabilités et les utilités indiquées sur l'arbre respecte les conditionnements imposés par le diagramme). Du fait que A_2 et U_1 sont indépendants de A_1 étant donné D_1 , les deux sous-arbres en pointillés sont identiques, de même que les deux sous-arbres en gras. Remarquons que la présence de sous-arbres identiques conduit à effectuer plusieurs fois les mêmes calculs lors de la détermination d'une stratégie maximisant EU dans l'arbre décision hasard. Les diagrammes d'influence permettent d'éviter cet écueil, ce que nous détaillons dans le paragraphe qui suit.

L'objet de notre travail est de "résoudre" un diagramme d'influence lorsque les préférences du décideur ne suivent pas le modèle EU mais le modèle RDU. Résoudre le diagramme consiste à déterminer la meilleure stratégie au sens du critère de décision utilisé (EU, RDU ou autre). Pour définir une stratégie, il est nécessaire de connaître les variables aléatoires déjà observées au moment de prendre chaque décision, ainsi que l'ordre chronologique dans lequel les dé-

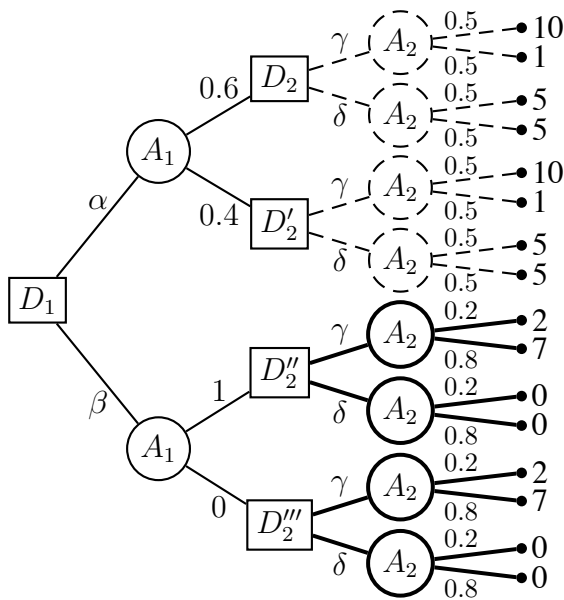


FIG. 2 – Arbre décision hasard.

cisions sont prises. Une stratégie consiste alors à fixer une valeur à chaque variable de décision conditionnellement à son passé. Dans un arbre décision hasard, le passé d’une variable de décision se définit simplement comme l’ensemble des variables aléatoires et des variables de décision situées sur le chemin allant de la racine à cette variable. L’ensemble des stratégies pour l’arbre de la figure 2 comporte 8 stratégies, dont en particulier la stratégie $\{D_1 = \alpha, D_2 = \gamma, D_2' = \delta\}$ (notons que les nœuds D_2'' et D_2''' ne peuvent être atteints lorsqu’on fixe $D_1 = \alpha$). Dans un diagramme d’influence, l’ordre chronologique est moins apparent. C’est pourquoi l’ensemble \mathcal{N}_A est partitionné en des ensembles disjoints I_0, I_1, \dots, I_n . L’ensemble I_0 comporte les variables aléatoires observées avant la première décision D_1 (correspondant aux parents de D_1), I_k les variables aléatoires observées entre D_k et D_{k+1} (correspondant aux parents de D_{k+1}), et enfin I_n les variables aléatoires restantes, c.à.d. celles qui ne sont jamais observées ou celles qui sont observées après la dernière décision D_n (i.e., les variables qui ne sont pas parents d’une variable de décision). Cela induit un ordre partiel \prec sur $\mathcal{N}_D \cup \mathcal{N}_A : I_0 \prec D_1 \prec I_1 \prec \dots \prec D_n \prec I_n$. Par exemple, pour le diagramme de la figure 1, l’ordre partiel est $D_1 \prec \{A_1\} \prec D_2 \prec \{A_2\}$. Le passé d’une variable de décision D_k se définit alors comme l’ensemble des variables X telles que $X \prec D$. Formellement, une stratégie dans un diagramme d’influence est un ensemble de règles de décision pour les variables D_k , où une règle de décision pour D_k

associe une valeur de son domaine pour chaque affectation des variables de son passé. Néanmoins, en pratique dans les diagrammes d’influence, seules les stratégies conséquentialistes (i.e., où chaque décision prise ne dépend que des variables influençant les paramètres futurs à la décision) sont envisagées. De ce fait, il n’est pas nécessaire de connaître les valeurs de toutes les variables du passé pour fixer la décision à prendre. Par exemple, pour le diagramme de la figure 1, du fait que A_2 et U_1 sont indépendants de A_1 étant donné D_1 , seule la variable D_1 influence les valeurs des variables A_2 et U_1 , qui constituent les paramètres décrivant le futur de D_2 . Comme indiqué en section 1, dans une stratégie conséquentialiste, la décision prise en D_2 ne dépend que des variables influençant les paramètres futurs (A_2 et U_1 ici), autrement dit la décision prise en D_2 ne dépend que de D_1 . Une règle de décision en D_2 s’écrira par exemple $D_2 = \gamma | D_1 = \alpha$ et $D_2 = \delta | D_1 = \beta$, ce qui signifie qu’on prend la décision γ (resp. δ) en D_2 si on a pris la décision α (resp. β) en D_1 . Par suite, l’ensemble des stratégies sur le diagramme de la figure 1 comporte les 4 stratégies suivantes : $\{D_1 = \alpha, D_2 = \gamma | D_1 = \alpha\}$ (notons que le cas $D_1 = \beta$ est omis dans la règle de décision en D_2 car on a fixé $D_1 = \alpha$), $\{D_1 = \alpha, D_2 = \delta | D_1 = \alpha\}$, $\{D_1 = \beta, D_2 = \gamma | D_1 = \beta\}$ et $\{D_1 = \beta, D_2 = \delta | D_1 = \beta\}$. Il est important de remarquer que l’ensemble des stratégies envisagées dans le diagramme d’influence n’est qu’un sous-ensemble des stratégies envisagées dans l’arbre décision hasard correspondant (par exemple la stratégie mentionnée plus haut pour l’arbre décision hasard ne figure pas dans ce sous-ensemble). Lorsqu’on optimise EU, cela ne porte pas à conséquence car il est bien connu qu’une stratégie EU-optimale se trouve dans ce sous-ensemble (puisque c’est une stratégie conséquentialiste). A contrario, une stratégie optimisant RDU ne se trouve pas nécessairement dans ce sous-ensemble (puisque ce n’est pas nécessairement une stratégie conséquentialiste), comme illustré par l’exemple suivant.

Exemple 3 *Considérons l’arbre décision hasard de la figure 2. Remarquons tout d’abord que toute stratégie dans laquelle $D_1 = \beta$ et $D_2'' = \delta$ ne peut être optimale au sens de RDU car toutes les conséquences possibles seraient alors nulles. De plus, on est indifférent à la valeur que peut prendre D_2''' puisque cette décision est conditionnée à l’occurrence d’un événement de probabilité nulle. Ainsi, seules 5 stratégies*

sont susceptibles de maximiser le critère RDU :

$$\begin{aligned} s_1 &= \{D_1 = \alpha, D_2 = \gamma, D'_2 = \gamma\}, \\ s_2 &= \{D_1 = \alpha, D_2 = \gamma, D'_2 = \delta\}, \\ s_3 &= \{D_1 = \alpha, D_2 = \delta, D'_2 = \gamma\}, \\ s_4 &= \{D_1 = \alpha, D_2 = \delta, D'_2 = \delta\}, \\ s_5 &= \{D_1 = \beta, D_2 = \gamma, D'_2 = \gamma\}. \end{aligned}$$

En supposant que les préférences du décideur suivent le modèle RDU avec la fonction φ définie par $\varphi(0) = 0$, $\varphi(p) = 0.45$ si $0 < p \leq 0.7$, $\varphi(p) = 1$ si $p > 0.7$, les valeurs selon RDU des différentes stratégies sont : $RDU(s_1) = RDU((0.5, 10; 0.5, 1)) = 1 + (10 - 1) \times \varphi(0.5) = 5.05$, $RDU(s_2) = 5.05$, $RDU(s_3) = 7.25$, $RDU(s_4) = 5$ et $RDU(s_5) = 7$. L'unique stratégie optimale selon RDU est s_3 , qui n'est pas conséquentialiste, puisqu'on prend deux décisions différentes en D_2 et D'_2 .

4 Conséquentialisme et cohérence dynamique

L'absence de conséquentialisme pourrait être vu comme un défaut du modèle RDU dans un contexte de décision séquentielle. L'exemple 4 ci-dessous est d'ailleurs souvent utilisé par les partisans de EU pour remettre en cause l'utilisation de RDU dans un contexte de décision séquentielle, car il semble montrer à première vue que le non-conséquentialisme conduirait à adopter des préférences qui ne sont pas *dynamiquement cohérentes*. Les préférences sont dites dynamiquement cohérentes lorsque les décisions (conditionnelles) futures jugées préférables aujourd'hui, à un nœud donné, continuent à l'être demain au nœud où elles devront être effectivement prises. Nous montrons par la suite que la critique de RDU vient d'une confusion entre cette dernière définition de la cohérence dynamique et le principe d'optimalité de la programmation dynamique ("toute sous-stratégie d'une stratégie optimale est optimale").

Exemple 4 *Considérons un candidat à un jeu télévisé qui a une chance sur dix de remporter le jeu au terme de l'émission. Si il l'emporte il peut repartir immédiatement avec 3000 € ou décider de retenter sa chance lors de l'émission du lendemain, et de repartir avec 4000 € s'il gagne à nouveau. Les statistiques indiquent qu'un candidat ayant gagné une première fois l'emporte neuf fois sur dix une seconde fois. Par soucis de commodité, nous représentons ce problème de décision séquentielle (élémentaire) à l'aide de l'arbre décision hasard de la figure 3,*

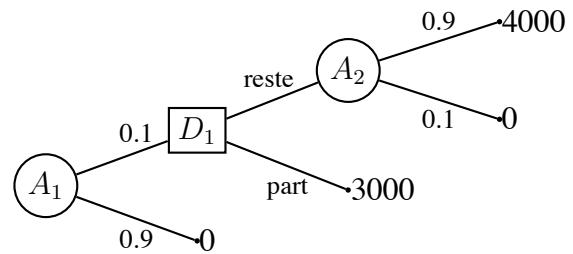


FIG. 3 – RDU et cohérence dynamique.

mais les arguments qui suivent resteraient valables dans le formalisme des diagrammes d'influence. Au nœud A_1 , la loterie obtenue pour la stratégie consistant à partir (resp. rester) après une première victoire est la loterie L_2 (resp. L'_2) de l'exemple 1. Au nœud D_1 , la loterie obtenue pour la stratégie consistant à partir (resp. rester) après une première victoire est la loterie L_1 (resp. L'_1) de l'exemple 1. Un candidat présentant le renversement de préférence fréquemment observé face à l'exemple de Allais aurait donc tendance à changer d'avis après avoir gagné une première fois (c'est-à-dire à partir finalement avec 3000 €, alors qu'il prévoyait initialement de viser les 4000 €).

A la suite de Machina [14], nous expliquons ici pourquoi cette volte-face n'aura pas lieu dès lors que le décideur est résolument non-conséquentialiste. Une différence importante entre un agent a dont les préférences suivent le modèle RDU (ou plus généralement un modèle autre que EU) et un agent b dont les préférences suivent le modèle EU vient en effet de ce que l'agent a accorde encore de l'importance à un risque *déjà encouru* (non-conséquentialisme), contrairement à b qui ne tient plus compte d'un risque une fois qu'il a été encouru. Autrement dit, un scénario contrefactuel joue un rôle dans RDU, contrairement à EU (un agent dont les préférences suivent le modèle EU a en effet un comportement conséquentialiste). Plus formellement, pour faire son choix en D_1 , l'agent a se conforme toujours aux préférences qu'il avait en A_1 , c'est-à-dire qu'en D_1 il compare toujours les loteries $(0.9, 0; 0.1, 3000)$ et $(0.91, 0; 0.09, 4000)$ (L_2 et L'_2), et non les loteries $(1, 3000)$ et $(0.9, 0; 0.1, 4000)$ (L_1 et L'_1). En explicitant les préférences ainsi, on se rend donc compte que l'agent reste cohérent dynamiquement car la sous-stratégie consistant à choisir la décision de rester une fois atteint le nœud D_1 reste préférée à celle de partir. Ce type de comportement a été formalisé en particulier par McClennen [15] sous le nom de *choix résolu*.

Le choix résolu désigne le comportement d'un agent qui se fixe un plan initialement et n'en dévie plus ensuite. Dans l'exemple 4, intuitivement, cela signifie que l'agent a , une fois établie son intention de participer pour viser les 4000 € ne changera pas d'avis avec la première victoire, afin de rester cohérent avec lui-même. Autrement dit, le modèle RDU vérifie bien la cohérence dynamique sous condition de choix résolu, même s'il ne vérifie pas le principe d'optimalité. C'est pourquoi nous nous intéressons dans la suite de cet article à déterminer un plan optimal au sens de RDU au vu de la situation initiale, et à s'y conformer ensuite. Remarquons que d'autres approches de choix résolu ont déjà été envisagées dans la littérature. Par exemple, Jaffray et Nielsen [9] considèrent chaque nœud de décision comme un *ego* de l'agent, et visent à déterminer une stratégie qui réalise un compromis entre les différents egos. Nous insistons donc sur le fait que ce n'est pas l'approche que nous adoptons ici, même si elle se justifie également.

5 Algorithmes

Du fait que la stratégie RDU-optimale n'est pas nécessairement conséquentialiste, une première méthode qui semble naturelle consiste à procéder en deux phases : 1) développer le diagramme d'influence en un arbre décision hasard, puis 2) déterminer la stratégie optimale au sens de RDU directement dans l'arbre décision hasard. Pour la phase 2, on peut s'appuyer sur un algorithme que nous avons proposé récemment [10] dans les arbres décision hasard (rappelons que la violation du principe d'optimalité empêche d'employer directement la programmation dynamique pour déterminer une stratégie optimale au sens de RDU). Toutefois, si elle présente l'avantage de permettre de déterminer la réelle stratégie optimale au sens de RDU, cette méthode est bien évidemment coûteuse en espace mémoire, et devient même impraticable lorsque la taille de l'arbre décision hasard est prohibitive.

C'est pourquoi nous proposons une autre méthode qui tire parti de la structure compacte des diagrammes d'influence (sans procéder à son développement en arbre décision hasard), au prix d'une réduction de l'ensemble des stratégies envisagées. Pour ce faire, une première idée serait d'explorer uniquement l'espace des stratégies conséquentialistes, mais nous perdriions alors une part importante des avantages descrip-

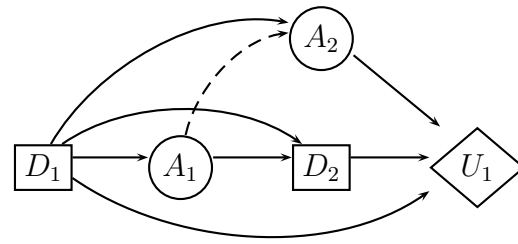


FIG. 4 – Diagramme d'influence modifié.

tifs du modèle RDU sur le modèle EU. C'est pourquoi l'approche que nous proposons ici ne renonce pas au non-conséquentialisme : nous introduisons une forme affaiblie de conséquentialisme afin de réaliser un compromis entre potentiel descriptif et compacité de la représentation. Il s'agit d'insérer des arcs fonctionnels supplémentaires dans le diagramme afin d'élargir l'espace (des stratégies) exploré. Remarquons que créer des dépendances supplémentaires factices entre variables indépendantes ne change pas le problème lui-même mais seulement sa représentation. Nous illustrons au travers de l'exemple 5 ci-dessous les changements induit par l'introduction d'un nouvel arc fonctionnel.

Exemple 5 *Après l'ajout de l'arc (A_1, A_2) sur le diagramme d'influence de la figure 1, on obtient le diagramme indiqué sur la figure 4. Du point de la perte de compacité de la représentation, il faut signaler que, si la variable A_1 est de modalité 2, cela aura pour effet de doubler le nombre de lignes dans la table assignée à la probabilité d'occurrence d'un événement de A_2 . En effet, la table initiale (avant ajout de l'arc) stockait la probabilité conditionnelle $P(A_2|D_1)$, tandis que la nouvelle table (après ajout de l'arc) représente $P(A_2|D_1, A_1)$. Du point de l'élargissement de l'espace des stratégies considérées, puisque A_1 influence les paramètres décrivant le futur de D_2 à la suite de l'ajout de l'arc, la décision prise en D_2 est maintenant conditionnée par la valeur de A_1 . Si la variable aléatoire A_1 peut prendre deux valeurs θ_1 et θ_2 , une règle de décision en D_2 s'écrira par exemple :*

$$\begin{aligned} D_2 &= \delta | D_1 = \alpha, A_1 = \theta_1, \\ D_2 &= \gamma | D_1 = \alpha, A_1 = \theta_2, \\ D_2 &= \gamma | D_1 = \beta, A_1 = \theta_1, \\ D_2 &= \delta | D_1 = \beta, A_1 = \theta_2. \end{aligned}$$

Autrement dit, dans ce cas précis, on constate que l'ajout de l'arc (A_1, A_2) permet de tenir compte de toutes les stratégies non-conséquentialistes.

Par la suite, nous appelons Δ -affaiblissement du conséquentialisme le fait d'ajouter Δ dépendances supplémentaires pour chaque variable de décision. Cela signifie qu'au pire on rajoute $|\mathcal{N}_D| \times \Delta$ arcs supplémentaires. L'ajout d'un arc dans le diagramme a cependant bien sûr une contrepartie : augmenter la taille d'une table du diagramme, et augmenter la masse de calculs réalisés par l'algorithme de résolution du diagramme. Pour limiter les effets négatifs liés à l'ajout d'un arc, on peut jouer sur deux paramètres :

- en tête de l'arc, la taille (i.e., le nombre de lignes) de la table assignée à la variable ;
- en queue de l'arc, la taille (i.e., le cardinal) du domaine de définition de la variable.

Par exemple, si l'on avait ajouté l'arc (A_1, U_1) sur le diagramme de la figure 1 plutôt que l'arc (A_1, A_2) , la décision en D_2 serait également devenue dépendante de A_1 . Cependant, le nombre de lignes de la table d'utilités en U_1 aurait été multiplié par 2, passant de 8 à 16 lignes, contre un passage de 2 à 4 lignes en A_2 pour l'arc (A_1, A_2) . Par ailleurs, il est important de remarquer que l'ajout d'un unique arc (V, V') , avec $V \in \mathcal{N}_A \cup \mathcal{N}_D$ et $V' \in \mathcal{N}_A \cup \mathcal{N}_U$, est susceptible de créer plusieurs dépendances $D_i|V$, où $V \prec D_i \prec V'$. Ainsi, il y a un arbitrage à réaliser entre le nombre d'arcs ajoutés et la taille des tables impliquées. Beaucoup d'heuristiques peuvent être envisagées pour accomplir cette tâche. Pour les expérimentations, nous avons développé une heuristique gloutonne ad-hoc pour les instances particulières générées, que nous explicitons dans la section portant sur les expérimentations numériques.

Nous allons maintenant détailler la procédure permettant de déterminer la stratégie maximisant le critère RDU à la suite d'un Δ -affaiblissement. Nous proposons ici une procédure par séparation et évaluation pour résoudre le diagramme d'influence.

Initialisation. L'initialisation consiste à déterminer une stratégie optimisant EU. Plusieurs algorithmes de résolution ont été proposés pour déterminer une stratégie maximisant EU dans un diagramme d'influence [20, 11]. L'algorithme de Shachter [20] consiste à éliminer incrémentalement les noeuds du diagramme d'influence tout en respectant l'ordre partiel sur ceux-ci. Pour ce faire, l'algorithme doit effectuer des inversions d'arcs de dépendance probabiliste, parfois très coûteuses en terme de calculs. L'algorithme de Jensen et al. [11] (inspiré des algorithmes d'inférence dans les *réseaux bayésiens*) consiste à transformer le dia-

gramme d'influence en un *arbre de jonction* et à faire de la programmation dynamique non sérielle [3] dans l'arbre de jonction obtenu. Ses performances étant généralement réputées supérieures à celle de l'algorithme de Shachter, nous avons adopté cette dernière approche dans notre implémentation.

Principe de branchement. Le principe de séparation que nous avons utilisé consiste simplement à fixer une valeur à une variable de décision donnée conditionnellement aux variables qui l'influencent. Par exemple, pour le diagramme de la figure 4, on sépare l'ensemble de stratégies caractérisé par :

$$(D_1 = \alpha) \wedge (D_2 = \gamma | D_1 = \alpha, A_1 = \theta_1)$$

en deux sous-ensembles caractérisés respectivement par :

$$\begin{aligned} & - (D_1 = \alpha) \wedge (D_2 = \gamma | D_1 = \alpha, A_1 = \theta_1) \\ & \quad \wedge (D_2 = \gamma | D_1 = \alpha, A_1 = \theta_2); \\ & - (D_1 = \alpha) \wedge (D_2 = \gamma | D_1 = \alpha, A_1 = \theta_1) \\ & \quad \wedge (D_2 = \delta | D_1 = \alpha, A_1 = \theta_2). \end{aligned}$$

Nous avons représenté l'arbre d'énumération sur la figure 6. Les nœuds instanciés y sont indiqués en gras. L'ordre d'instanciation des variables dans l'arbre d'énumération est compatible avec l'ordre total sur les nœuds de décision (rappelons qu'il existe toujours un chemin reliant tous les nœuds de décision du diagramme).

Calcul de la borne inférieure. En chaque nœud de l'arbre d'énumération, on calcule une stratégie optimisant EU dans le sous-ensemble des stratégies considérées. Lorsque sa valeur selon RDU est plus grande que la valeur de la meilleure stratégie trouvée jusqu'alors, elle prend la place de cette dernière.

Calcul de la borne supérieure. Comme nous l'avons précisé, à chaque stratégie on peut associer une loterie (variable aléatoire sur les conséquences potentielles) et c'est l'évaluation de ces loteries par un critère de décision qui permet de comparer les stratégies. Nous basons notre évaluation sur la notion de dominance stochastique entre les loteries (rappelons qu'une loterie L domine stochastiquement une loterie L' si $\forall x \in \mathbb{R}, G_L(x) \geq G_{L'}(x)$). Il s'agit de calculer une loterie qui domine stochastiquement toutes les stratégies potentielles compatibles avec les décisions déjà fixées, son évaluation selon RDU fournissant une borne supérieure (par compatibilité de RDU avec la dominance stochastique, i.e. $RDU(L) \geq RDU(L')$ dès lors que L domine L'). Le calcul peut se faire par programmation dynamique, en déterminant récursivement une telle loterie pour chaque variable de décision et chaque instanciation possible des va-

riables de son passé (compatible avec les décisions déjà fixées). Dans la suite, pour simplifier, on suppose qu'il y a une seule variable d'utilité U à valeurs dans $\{u_1, \dots, u_m\}$ avec $u_1 \leq \dots \leq u_m$. On procède par induction arrière, l'initialisation en U étant réalisée par la formule suivante :

$$\forall i \geq 1, \quad P(U = u_i | I_0 \cdot \dots \cdot I_n, D_1 \cdot \dots \cdot D_n) \\ = \begin{cases} 1 & \text{si } U(I_0 \cdot \dots \cdot I_n, D_1 \cdot \dots \cdot D_n) = u_i \\ 0 & \text{sinon} \end{cases}$$

où $U(I_0 \cdot \dots \cdot I_n, D_1 \cdot \dots \cdot D_n)$ indique la valeur que prend la variable U en fonction des valeurs des variables $I_0, \dots, I_n, D_1, \dots, D_n$. Considérons maintenant une variable de décision D_k . Pour une réalisation donnée des variables aléatoires I_0, \dots, I_{k-1} et un choix donné de décisions en D_1, \dots, D_{k-1} , la loterie dominante en D_k se calcule par l'équation de récurrence suivante :

$$\forall i \geq 1, \quad P(U \geq u_i | I_0 \cdot \dots \cdot I_{k-1}, D_1 \cdot \dots \cdot D_{k-1}) \\ = \max_{D_k} \sum_{j=i}^m \sum_{I_k} P(I_k | I_0 \cdot \dots \cdot I_{k-1}, D_1 \cdot \dots \cdot D_k) \\ * P(U = u_j | I_0 \cdot \dots \cdot I_{k-1}, I_k, D_1 \cdot \dots \cdot D_k)$$

La loterie dominante proprement dite se déduit facilement via la formule :

$$P(U = u_i | I_0 \cdot \dots \cdot I_{k-1}, D_1 \cdot \dots \cdot D_{k-1}) \\ = P(U \geq u_i | I_0 \cdot \dots \cdot I_{k-1}, D_1 \cdot \dots \cdot D_{k-1}) \\ - P(U \geq u_{i+1} | I_0 \cdot \dots \cdot I_{k-1}, D_1 \cdot \dots \cdot D_{k-1})$$

La valeur retournée au final est calculée en appliquant RDU à la loterie obtenue en D_1 . Notons ici qu'en pratique les conditionnements se font sur des ensembles beaucoup plus restreints de variables grâce aux multiples indépendances mises en avant dans le diagramme d'influence. Enfin, précisons que, pour optimiser les calculs, nous avons utilisé, à la manière de l'algorithme de Jensen [11], un arbre de jonction (construit à partir du diagramme d'influence).

Exemple 6 *Considérons à nouveau l'arbre d'énumération représenté sur la figure 6. Il y a 2 stratégies possibles compatibles avec les règles de décisions déjà fixées, dont les loteries associées sont $L_1 = (1, 0.5; 10, 0.5)$ et $L_2 = (1, 0.3; 5, 0.4; 10, 0.3)$. Nous indiquons sur le repère de gauche de la figure 5 leurs fonctions décumulatives. La loterie dominante calculée par programmation dynamique correspondra alors à la loterie dont la fonction décumulative est le max de G_{L_1} et G_{L_2} . Nous indiquons cette fonction décumulative en gras sur le repère de droite de la figure 5. Cette loterie dominante est ensuite évaluée selon le critère RDU et nous fournit une borne supérieure des valeurs RDU des*

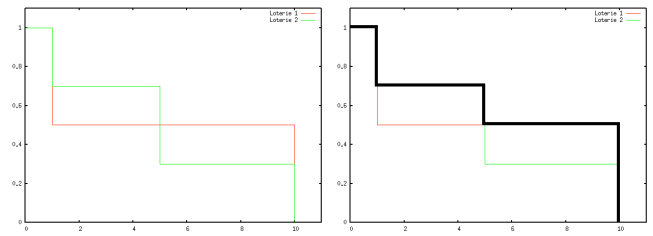


FIG. 5 – Exemple du calcul d'une loterie dominante

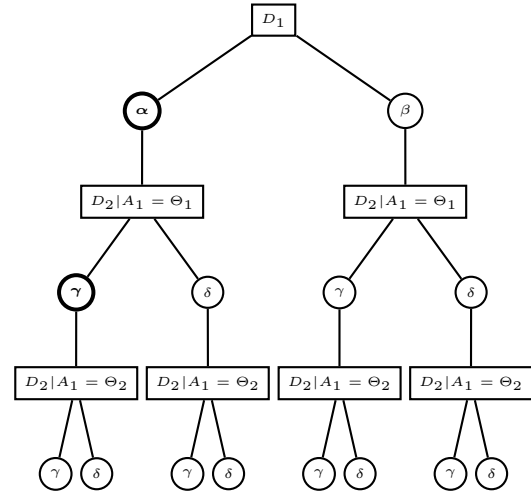


FIG. 6 – Arbre d'énumération.

stratégies compatibles avec les règles de décisions fixées dans l'arbre d'énumération.

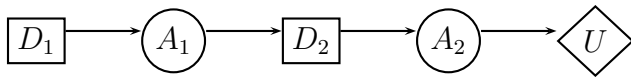
6 Expérimentations numériques

Les deux algorithmes proposés (la méthode en deux phases et celle opérant directement sur le diagramme d'influence enrichi de dépendances fictives) ont été implémentés en C++ et les tests numériques ont été menés sur un ordinateur muni d'un processeur Pentium IV à 2.13Ghz et 3.5Go de mémoire vive. Nous présentons ci-dessous les résultats de tests effectués sur des diagrammes d'influence générés aléatoirement. Nous décrivons tout d'abord le fonctionnement du générateur aléatoire, avant de détailler les résultats obtenus, tant au niveau des performances en temps que de la qualité des stratégies retournées. Pour toutes les expérimentations, nous avons utilisé la fonction φ de déformation suivante : $\varphi(p) = p^\gamma / (p^\gamma + (1-p)^\gamma)$. Cette fonction de déformation est celle proposée usuellement

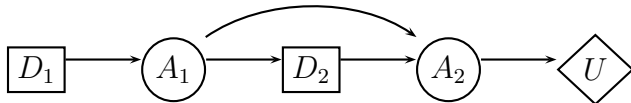
dans la littérature [18]. Le paramètre γ est à valeur dans $[0, 1]$. Remarquons que, pour $\gamma = 1$, on a $\varphi(p) = p$ et on retrouve alors le cas particulier EU. Nous avons fixé ici $\gamma = 0.5$.

6.1 Génération d'instances

Afin de réellement contrôler la taille des diagrammes d'influence générés, il est nécessaire que tous les nœuds du diagramme entrent en compte lors du calcul de la stratégie optimale. Pour ce faire, étant donné un nombre n fixé de variables de décision, on crée en premier lieu un chemin de longueur $2n+1$ en alternant variables de décisions et variables aléatoires, pour terminer sur une variable d'utilité. Par exemple, pour 2 variables de décision on obtient le chemin suivant :



Ensuite, pour éviter que certaines variables du diagramme d'influence ne jouent aucun rôle réel (i.e., n'aient pas d'impact sur le choix de la stratégie optimale), on impose que chaque variable aléatoire influence une autre variable aléatoire (dès lors qu'il en existe au moins une dans son futur). En reprenant l'exemple précédent on peut alors obtenir :



Enfin, on insère des arcs supplémentaires de manière aléatoire entre certains nœuds.

En ce qui concerne le Δ -affaiblissement par ajout de Δ dépendances supplémentaires pour chaque variable de décision, nous avons procédé comme suit. Pour chaque variable de décision D_k , on rajoute un arc dont le sommet de départ est le nœud précédant D_k et dont le sommet d'arrivée est le nœud (représentant une variable aléatoire ou une variable de décision) de plus petit degré entrant dans le futur de D_k . Ensuite, on réitère jusqu'à ajouter Δ arcs en prenant à chaque fois le sommet N de départ un rang plus tôt dans le passé (dans l'ordre total induit par notre mode de génération des instances), sous condition que D_k ne dépende pas déjà de N .

6.2 Résultats obtenus

Afin d'évaluer le gain computationnel réalisé lorsque l'on travaille directement sur le diagramme d'influence, nous avons comparé les temps de résolution pour la méthode en deux phases avec les temps de résolution pour différentes valeurs de Δ . Le tableau 2 indique ces temps d'exécutions (temps moyens en seconde sur l'ensemble des instances testées). Dans chaque colonne on fait varier le nombre de variables de décision (i.e. on fait varier la valeur de n) et dans chaque ligne on fait varier la valeur de Δ . Dans la dernière ligne du tableau figure le temps mis par la méthode en deux phases. Pour chaque valeur de n , 40 instances ont été générées aléatoirement pour lesquelles on a appliqué successivement l'approche par Δ -affaiblissement pour différentes valeurs de Δ et l'approche en deux phases. On constate sans surprise que plus la taille du diagramme et la valeur de Δ augmentent, plus le temps de résolution augmente. La méthode en deux phases ne permet pas d'aller au delà de 7 nœuds de décision. La méthode par Δ -affaiblissement permet de traiter des instances de taille plus importantes, au prix de l'optimalité. Nous nous intéressons précisément dans le paragraphe suivant à la qualité des solutions retournées.

$\Delta \backslash n$	4	5	6	7	8	9	10
0	0	0	0	0.21	0.85	2.78	9.11
1	0	0	0	0.23	0.62	3.94	12.15
2	0	0	0	0.38	1.10	4.79	23.36
3	0	0	0.10	0.72	2.33	9.25	29.97
4	0	0	0.56	1.07	4.46	18.61	93.52
5	0	0.08	1.91	3.43	19.84	87.40	—
2 ph.	0.17	0.93	5.02	17.38	—	—	—

TAB. 2 – Temps d'exécution (sec.).

Remarquons que la stratégie optimale au sens de EU peut être vue comme une approximation de la stratégie optimale au sens de RDU. Cette approximation peut être calculée rapidement en utilisant les algorithmes classiques dans les diagrammes d'influence. Etant donné un diagramme d'influence DI , notons $RDU_{EU}(DI)$ (resp. $RDU_{\Delta}(DI)$) la valeur RDU de la stratégie optimisant EU (resp. optimisant RDU dans DI au terme d'un Δ -affaiblissement du conséquentialisme). Afin d'évaluer l'influence du paramètre Δ sur la qualité des stratégies obtenues au sens de RDU, nous avons calculé le ratio $(RDU_{\Delta}(DI) - RDU_{EU}(DI))/RDU_{EU}(DI)$ pour différentes valeurs de Δ et du nombre n de variables de décision. Pour chaque couple

(Δ, n) , 200 instances *DI* ont été aléatoirement générées. Les résultats obtenus sont indiqués dans le tableau 3. Dans chaque case du tableau on indique le ratio moyen (exprimé en pourcentage) sur les 200 instances, ainsi que le plus grand ratio obtenu. Dans la dernière ligne du tableau, on indique les résultats toutes valeurs de n confondues. On remarque que le ratio moyen augmente significativement lorsque Δ croît. Par ailleurs, le gain obtenu est monté jusqu'à 48% sur certaines instances.

n	$\Delta = 2$		$\Delta = 3$		$\Delta = 4$		$\Delta = 5$	
	moy	max	moy	max	moy	max	moy	max
2	11%	26%	—	—	—	—	—	—
3	6%	9%	6%	11%	13%	32%	—	—
4	3%	6%	7%	23%	18%	48%	14%	36%
5	3%	8%	4%	13%	10%	23%	14%	40%
6	1%	11%	9%	27%	7%	28%	11%	30%
	5%	26%	6%	27%	12%	48%	13%	40%

TAB. 3 – Influence du paramètre Δ sur la qualité des stratégies obtenues.

7 Conclusion

Dans cet article, nous avons proposé une approche algorithmique pour optimiser RDU dans un diagramme d'influence, en visant à concilier des préoccupations d'optimalité (obtenir une stratégie dont la valeur RDU soit la plus proche possible de l'optimum) et de compacité de représentation (obtenir une stratégie dont le stockage en machine ne soit pas trop gourmand en mémoire). Nous avons montré que l'espace des stratégies habituellement envisagées dans les diagrammes d'influence (espace des stratégies dites *conséquentialistes*) est bien plus restreint que l'espace des stratégies potentielles dans l'arbre décision hasard obtenu par développement du diagramme (espace qui inclut bien sûr le sous-espace précédent mais aussi l'ensemble des stratégies non-conséquentialistes). C'est pourquoi, dans notre approche, le diagramme d'influence est muni d'arcs supplémentaires représentant des dépendances fictives entre décision afin d'enrichir l'espace exploré par un sous-ensemble de stratégies non-conséquentialistes (Δ -affaiblissement du conséquentialisme, qui correspond à l'ajout de Δ arcs fictifs par nœud de décision). Les expérimentations numériques menées montrent l'intérêt de procéder à ce Δ -affaiblissement car la valeur RDU de la stratégie calculée améliore parfois considérablement la valeur obtenue pour une stratégie conséquentialiste.

Afin d'améliorer l'approche proposée ici, la piste de recherche la plus prometteuse nous semble être un travail sur l'identification des arcs dont l'ajout dans le diagramme d'influence joue le plus fortement sur la valeur RDU de la stratégie retournée. En effet, cela permettrait à la fois de limiter la croissance de la taille du diagramme (liée à l'ajout d'arcs) et d'améliorer la qualité de la stratégie retournée. Dans une toute autre direction, on pourrait étudier l'utilisation de RDU dans les processus décisionnels markoviens. La difficulté dans ce nouveau cadre est que la stratégie optimisant RDU à horizon infini n'est pas nécessairement markovienne. Un affaiblissement de la propriété markovienne similaire à l'affaiblissement du conséquentialisme présenté ici pourrait être envisagé.

Références

- [1] M. Allais, Le comportement de l'homme rationnel devant le risque : critique des postulats de l'école américaine, *Econometrica*, Vol. 21, pp. 503-546, 1953.
- [2] M. Allais, An outline of my main contributions to economic science, *The American Economic Review*, Vol. 87, Num. 6, pp. 3-12, 1997.
- [3] U. Bertelé et F. Brioschi, *Nonserial Dynamic Programming*, Academic Press, 1972.
- [4] T. Dean, Planning with deadlines in stochastic domains, *Proc. of the 11th AAAI*, pp. 574-579, 1993.
- [5] J.-P. Gayant, *Risque et décision*, Vuibert, 2001.
- [6] J. Handa, Risk, probabilities and a new theory of cardinal utility, *Journal of Political Economics*, Vol. 85, pp. 97-122, 1977.
- [7] R. Howard et J. Matheson, Influence Diagrams, *Readings on the Principles and Applications of Decision Analysis*, pp. 721-762, 1984.
- [8] J.-Y. Jaffray, Décision séquentielle et rationalité, *Actes de PDMIA*, 2004.
- [9] J.-Y. Jaffray et T. Nielsen, An operational approach to rational decision making based on rank dependent utility, *European Journal of Operational Research*, Vol. 169, Num. 1, pp. 226-246, 2006.
- [10] G. Jeantet et O. Spanjaard, Rank-Dependent Probability Weighting in Sequential Decision Problems under Uncertainty, *Proc. of ICAPS*, pp. 148-155, 2008.
- [11] F. Jensen, F.V. Jensen et S.L. Dittmer, From Influence Diagrams to Junction Trees, *UAI 1994*, pp. 367-373, 1994.
- [12] D. Kahneman et A. Tversky, Prospect theory : An analysis of decision under risk, *Econometrica*, Vol 47, pp. 263-291, 1979.
- [13] Y. Liu et S. Koenig, An exact algorithm for solving MDPs under risk-sensitive planning objectives with one-switch utility functions, *Proc. of the Int. Joint Conf. on Autonomous Agents and Multiagent Systems (AAMAS)*, pp. 453-460, 2008.

- [14] M.J. Machina, Dynamic Consistency and Non-Expected Utility Models of Choice Under Uncertainty, *Journal of Economic Literature*, pp. 1622-1668, 1989.
- [15] E. McClennen, *Rationality and Dynamic choice : Foundational Explorations*, Cambridge University Press, 1990.
- [16] P. Perny, O. Spanjaard et L.-X. Storme, State space search for risk-averse agents, *20th International Joint Conference on Artificial Intelligence*, pp. 2353-2358, 2007.
- [17] J. Quiggin, A theory of anticipated utility, *Journal of Economic Behavior and Organisation*, Vol. 3, Num. 4, pp. 323-343, 1982.
- [18] J. Quiggin, *Generalized Expected Utility Theory : The Rank-Dependent Model*, Kluwer, 1993.
- [19] H. Raiffa, *Decision Analysis : Introductory Lectures on Choices under Uncertainty*, Addison-Wesley, 1968.
- [20] R. Shachter, Evaluating influence diagrams, *Operations Research*, Vol. 34, pp. 871-882, 1986.
- [21] J. von Neuman et O. Morgenstern, *Theory of games and economic behaviour*, Princeton University Press, 1947.