THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par : *Institut Supérieur de l'Aéronautique et de l'Espace (ISAE)*

NICOLAS DROUGARD

Tirer Profit de Sources d'Information Imprécises pour la Décision Séquentielle dans l'Incertain

---

JURY

DIDIER DUBOIS	Directeur de recherche CNRS	Membre
JEAN-LOUP FARGES	Ingénieur de recherche	Membre
HECTOR GEFFNER	Professeur	Rapporteur
PATRICE PERNY	Professeur	Rapporteur
FLORENT TEICHTEIL-KÖNIGSBUCH	Ingénieur de recherche	Membre
BRUNO ZANUTTINI	Maître de conférence, HDR	Membre

École doctorale et spécialité :
EDSYS : Systèmes embarqués
Unité de Recherche :
Onera – The French Aerospace Lab, département DCSD, unité CD
Directeur(s) de Thèse :
Didier Dubois, *Florent Teichteil-Königsbuch* et *Jean-Loup Farges*
Rapporteurs :
Hector Geffner et *Patrice Perny*

Ce manuscrit est un résumé en français qui représentant un peu plus d'un quart du travail original, qui lui est en anglais.

Table des matières

Liste des figures

Introduction

Contexte

L'autonomie décisionnelle d'un robot provient, entre autre, du calcul d'une fonction appelée *stratégie* ou *politique* : celle-ci retourne le symbole de l'action à exécuter en fonction de l'historique des données des capteurs du robot. Les caractéristiques d'intérêt du robot et de son environnement forment un *système*. En général, pour une séquence donnée d'actions exécutées par le robot, l'évolution de ce système n'est pas déterministe. Cependant le comportement du système robotique peut être étudié en effectuant de nombreux tests sur lui *i.e.* des exécutions d'actions, ou en utilisant les connaissances d'un expert de ce système. De même, les données provenant des capteurs, brutes ou traitées, sont généralement des éléments incertains : le comportement de ces données, appelées aussi *observations* du système, dépend des actions du robot et des états successifs du système. Des relations peuvent aussi être obtenues entre les observations, les états du système et les actions à l'aide de tests des capteurs dans de nombreuses situations, de la description technique de ces même capteurs, du traitement des données utilisé, ou de n'importe quelle information experte. Par exemple, dans le cas d'un robot utilisant une vision artificielle, la sortie de l'algorithme de traitement d'image utilisé est considérée comme une observation du système puisque elle est à la fois le résultat d'un traitement des données des capteurs, et l'information sur laquelle se base le modèle de décision : ici, les données sont les images provenant de la camera. Pour une camera donnée, et un algorithme de vision donné, le comportement de l'observation est lié à l'action et à l'état du système lors du processus de prise d'image.

Ainsi, dans le but de rendre un robot autonome pour une *mission* donnée, nous cherchons une stratégie, *i.e.* une fonction précisant les actions à exécuter conditionnellement à la séquence d'observations du système, qui tient compte de l'incertitude à propos de l'évolution du système et de ses observations. Le domaine de recherche associé à ce type de problème, *i.e.* le problème du calcul de stratégies, n'est pas restreint à la robotique et est appelé *décision séquentielle dans l'incertain* : dans le cas général, l'entité qui doit effectuer l'action est appelée *l'agent*. Dans cette thèse, bien que les résultats fournis sont principalement théoriques et assez généraux pour concerner des applications plus variées, le problème du calcul de stratégie est étudié ici dans le contexte de la robotique autonome, et l'agent est la partie décisionnelle du robot. Calculer une stratégie pour une mission robotique donnée nécessite un cadre adapté : le modèle le plus connu décrit le comportement de l'état du système et des observations en utilisant la théorie de probabilités.

Un modèle probabiliste pour le calcul de stratégies

Les Processus Décisionnels de Markov (PDM) expriment aisément les problèmes de décision séquentielle sous incertitude probabiliste [5]. Ils sont adaptés au calcul de stratégies si l'état du système est connu par l'agent à chaque moment de la mission. Dans le contexte robotique, cette hypothèse signifie que dans la mission considérée, le robot a une connaissance parfaite des caractéristiques d'intérêt du problème via ses capteurs. Dans ce modèle, l'état du système

est noté s, et l'ensemble fini de tous les états possibles du système est noté $\mathcal{S}$. L'ensemble fini $\mathcal{A}$ est l'ensemble de toutes les actions disponibles pour l'agent qui sont notées $a \in \mathcal{A}$. Le temps est discrétisé en étapes de décision représentées par les entiers $t \in \mathbb{N}$.

Il est supposé que la dynamique de l'état du système est *markovienne* : à chaque étape de temps t, l'état suivant $s_{t+1} \in \mathcal{S}$, ne dépend que de l'état courant $s_t \in \mathcal{S}$ et de l'action choisie $a_t \in \mathcal{A}$. Cette relation est décrite par la fonction de transition $\mathbf{p}(s_{t+1} \mid s_t, a_t)$, définie comme la distribution de probabilité sur l'état suivant s_{t+1} conditionnellement à l'état courant $s_t \in \mathcal{S}$ lors de l'exécution de l'action $a_t \in \mathcal{A}$.

La mission de l'agent est décrite en termes de *récompenses*. Une fonction de récompense $r : (s, a) \mapsto r(s, a) \in \mathbb{R}$ est définie pour chaque action $a \in \mathcal{A}$ et chaque état du système $s \in \mathcal{S}$. Elle modélise le but de l'agent. Chaque valeur de récompense $r(s, a) \in \mathbb{R}$ est une motivation locale pour l'agent. En effet, plus l'agent récupère de récompenses pendant l'exécution du processus, mieux il a réalisé sa mission : une mission est considérée bien remplie si la séquence d'états du système $s_t \in \mathcal{S}$ rencontrés et d'actions $a_t \in \mathcal{A}$ sélectionnées mènent à des récompenses $r(s_t, a_t)$ grandes. La résolution d'un PDM correspond au calcul d'une stratégie optimale, *i.e.* d'une fonction prescrivant les actions $a \in \mathcal{A}$ qu'il faut exécuter au cours du temps afin de maximiser la moyenne de la somme des récompenses obtenues durant une exécution : cette moyenne est calculée en tenant compte du comportement probabiliste des états du système décrit par les fonctions de transition $\mathbf{p}(s_{t+1} \mid s_t, a_t)$. Par exemple, une bonne stratégie peut être une fonction d définie sur $\mathcal{S}$ et à valeurs dans $\mathcal{A}$, puisque l'on suppose ici que l'agent connaît l'état du système durant le processus.

Il a été montré que de telles stratégies markoviennes sont optimales pour certain critères tels que celui basé sur la somme des récompenses actualisées : en effet, un critère bien connu mesurant les performances d'une stratégie d est l'espérance de la somme actualisée des récompenses :

$$\mathbb{E}\left[\sum_{t=0}^{+\infty} \gamma^t r(s_t, d_t)\right], \tag{1}$$

où $d_t = d(s_t) \in \mathcal{A}$ et $0 < \gamma < 1$ est un facteur d'actualisation assurant la convergence de la somme.

L'hypothèse que l'agent a une connaissance parfaite de l'état du système est assez forte : en particulier, dans le cas des robots réalisant des taches avec des capteurs conventionnels, ces derniers sont souvent incapable de fournir au robot toutes les caractéristiques d'intérêt pour la mission. Ainsi, un modèle plus flexible a été construit : il tient compte de *l'observation partielle* du système par l'agent.

Les PDM Partiellement Observables (PDMPO) [70] ont une puissance de modélisation plus importante, car ils peuvent représenter des situations dans lesquelles l'agent ne connaît pas directement l'état courant du système : ils modélisent de manière plus fine un agent exécutant des actions sous incertitude dans un environnement partiellement observable.

L'ensemble des états du système $\mathcal{S}$, l'ensemble des actions $\mathcal{A}$, la fonction de transition $\mathbf{p}(s_{t+1} \mid s_t, a_t)$ et la fonction de récompense $r(s, a)$ restent les mêmes que pour la définition des PDM. Dans ce modèle, puisque l'état courant du système $s \in \mathcal{S}$ ne peut pas être considéré comme une information accessible pour l'agent, la connaissance de l'agent à propos de l'état du système provient des observations $o \in \mathcal{O}$, où $\mathcal{O}$ est un ensemble fini. La fonction d'observation $\mathbf{p}(o_{t+1} \mid s_{t+1}, a_t)$ donne pour chaque action $a_t \in \mathcal{A}$ et état atteint $s_{t+1} \in \mathcal{S}$, la probabilité sur les observations possibles $o_{t+1} \in \mathcal{O}$. Enfin, *l'état de croyance initial* $b_0(s)$ définit la distribution de probabilité *a priori* sur l'état du système. Un exemple d'usage des PDMPO est illustré dans la figure 1.

Résoudre un PDMPO consiste à calculer une stratégie qui renvoie une action adéquate à chaque étape du processus, et dépendente des observations reçues et des actions sélectionnées

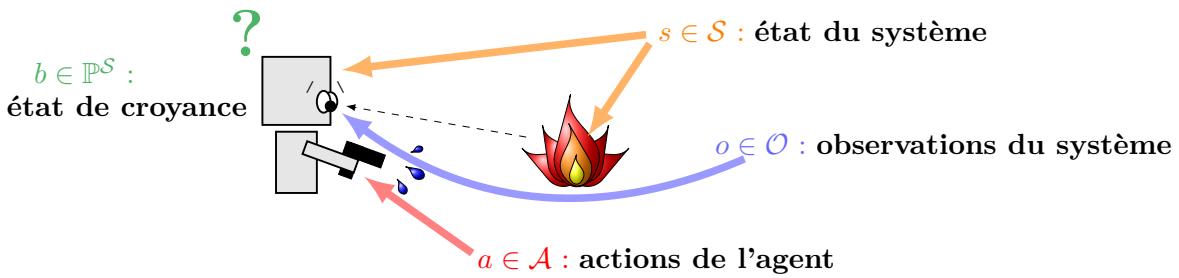


FIGURE 1 – *Utilisation d'un PDMPO pour la modélisation du robot pompier : dans cet exemple, la mission du robot est la prévention des incendies. L'état du système* $s \in \mathcal{S}$ *décrit par exemple la position du robot, l'orientation du jet d'eau, la quantité d'eau utilisée, la position du feu et son niveau sur une échelle de "petit feu" à "feu important", etc. En utilisant une vision artificielle et des capteurs de chaleur, le robot reçoit des* ***observations*** $o \in \mathcal{O}$ *qui sont les données brutes ou traitées provenant des capteurs : la sortie d'un classifieur dont l'entrée est une image de la scène (cf. Figure 3), et qui renvoie une estimation du niveau ou de la position du feu, peut être modélisée par une observation. Finallement, les* ***actions du robot*** $a \in \mathcal{A}$ *sont, par exemple, la mise en marche des moteurs impactant la rotation des roues du robot, le débit de pompage, l'orientation du jet d'eau ou des capteurs, etc. La* ***fonction de récompense*** $r(s,a)$ *décroît avec le niveau de l'incendie. Afin de ne pas gaspiller d'eau, un coût proportionnel à la quantité d'eau est soustrait à cette récompense : puisque une stratégie optimale maximise la moyenne de la somme des récompenses, le but du robot est donc d'éteindre les incendies sans gaspiller de l'eau. Cette moyenne peut-être calculée à l'aide des probabilités décrivant la dynamique stochastique du système. Les actions du robot* $a \in \mathcal{A}$ *ont un effet probabiliste sur le système, décrit par la* ***fonction de transition*** $\boldsymbol{p}(s' \mid s,a)$ *: par exemple, l'activation des roues du moteur modifie la position du robot, et la probabilité sur chacune des positions suivantes possibles, étant donnée la position courante, prend part à la définition du PDMPO. Un autre exemple est l'action modifiant l'orientation du jet d'eau, qui redéfinit la probabilité du nouveau niveau de feu étant donné l'état actuel du système. Les actions du robot* $a \in \mathcal{A}$ *et les états suivants* $s' \in \mathcal{S}$ *peuvent aussi impacter les observations des capteurs : cette influence est définie par la* ***fonction d'observation*** $\boldsymbol{p}(o' \mid s',a)$*, où* $o' \in \mathcal{O}$ *est l'observation : par exemple, l'orientation du capteur de vision peut modifier la probabilité de détection du feu, ou de l'évaluation de son intensité, qui font partie des observations* $o' \in \mathcal{O}$*. Finalement, l'état de croyance est la distribution de probabilité sur l'état courant du système conditionnellement à l'ensemble des observations et des actions successives depuis le début du processus : c'est la meilleure estimation possible puisque le robot n'a accès qu'aux actions et observations lors de l'exécution.*

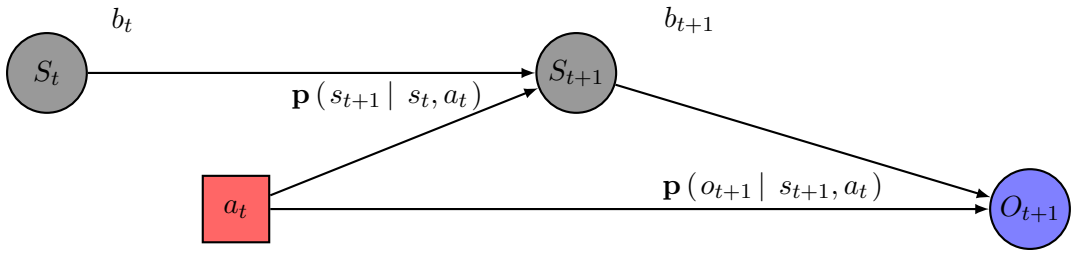


FIGURE 2 – *Réseau Bayésien illustrant la mise à jour de l'état de croyance : les états sont représentés par des ronds gris, l'action est représentée par le losange rouge, et l'observation par le rond bleu. La variable aléatoire S_{t+1} représentant l'état suivant s_{t+1} dépend de l'état courant s_t et de l'action courante a_t. La variable aléatoire O_{t+1} réprésentant l'observation suivante o_{t+1} dépend de l'état suivant s_{t+1} et de l'action courante a_t. L'état de croyance b_t (resp. b_{t+1}) est l'estimation probabiliste de l'état courant (resp. suivant) du système, s_t (resp. s_{t+1}).*

i.e. de toutes les données disponibles pour l'agent : un critère courant pour la stratégie est l'espérance de la somme actualisée des récompenses (1).

La plupart des algorithmes raisonnent sur *l'état de croyance*, défini comme la distribution de probabilité sur l'état du système conditionnellement à toutes les observations du système et les actions choisies par l'agent depuis le début du processus. Cet état de croyance est mis à jour à chaque étape de temps en utilisant la règle de Bayes, l'action courante, et la nouvelle observation. À une étape donnée $t \in \mathbb{N}$, $b_t(s)$ est défini comme la probabilité que le $t^{ième}$ état du système soit $s \in \mathcal{S}$, connaissant les observations et actions précédentes, ainsi que l'état de croyance initial b_0 : c'est une estimation de l'état du système qui utilise uniquement les données disponibles puisque l'état n'est pas directement observable.

Il peut être facilement calculé de manière récursive avec la règle de Bayes : à l'étape de temps t, si l'état de croyance est b_t, l'action choisie $a_t \in \mathcal{A}$ et la nouvelle observation $o_{t+1} \in \mathcal{O}$, l'état de croyance suivant est

$$b_{t+1}(s') \propto \mathbf{p}(o_{t+1} \mid s', a_t) \cdot \sum_{s \in \mathcal{S}} \mathbf{p}(s' \mid s, a_t) \cdot b_t(s). \tag{2}$$

comme illustré par le réseau bayésien de la figure 2.

Puisque les états de croyance successifs sont calculés avec les observations perçues par l'agent, ils sont considérés visible par l'agent. Notons $\mathbb{P}^{\mathcal{S}}$ l'ensemble continu des distributions de probabilité sur $\mathcal{S}$. Une stratégie optimale peut être cherchée parmi les fonctions d définies sur $\mathbb{P}^{\mathcal{S}}$ telles que les $d_t = d(b_t) \in \mathcal{A}$ successifs maximisent l'espérance des récompenses (1) : les décisions de l'agent sont alors basées sur l'état de croyance.

Les PDMPO fournissent un cadre flexible pour la robotique autonome, comme illustré par l'exemple du robot pompier, *cf.* figure 1 : ils permettent de décrire le système regroupant le robot et son environnement, ainsi que la mission du robot. Ils sont assez fréquemment utilisés en robotique [58, 52, 48, 15, 16]. En effet, ils prennent en compte le fait que le robot ne reçoit que les données des capteurs, et doit estimer l'état du système (qui lui est caché) afin de réaliser sa mission. Pour cela, il utilise ces données, appelées alors observations. Cependant, le modèle PDMPO soulève quelques problèmes, en particulier dans le contexte robotique.

PROBLÈMES PRATIQUES DES PDMPO

Complexité

Résoudre un PDMPO *i.e.* calculer une stratégie optimale, est PSPACE-hard en horizon fini [54] et même indécidable en horizon infini [47]. De plus, un espace exponentiel en la description

FIGURE 3 – *Exemple d'une méthode d'observation dans le contexte robotique : le robot, ici un drone, est équippé d'une caméra et utilise un classifieur (classifier) calculé à partir d'une base de données d'images (comme NORB, cf. figure 4). Le classifieur est généré avant la mission (hors-ligne, off-line) avec une base de données d'images (cf. partie droite de l'illustration, dataset), et la sortie du classifieur est utilisée lors de la mission (en ligne, online) comme une observation pour l'agent (cf. partie gauche de l'illustration). Ici, les observations sont donc générées par un algorithme de vision artificielle.*

du problème peut être requis pour la spécification explicite d'une telle stratégie. Le travail [49] est un bon résumé des l'analyses de complexité des PDMPO.

Cette forte complexité est très bien connue des utilisateurs des PDMPO : l'optimalité ne peut être atteinte que pour des petits problèmes, ou bien des problèmes très structurés. Les approches classiques essaient de résoudre ce problème en utilisant la programmation dynamique et des techniques de programmation linéaire [14]. Cependant, pour des problèmes non triviaux, seules des solutions approchées peuvent être calculées, et donc la stratégie n'a pas de garantie d'optimalité. Par exemple, les approches populaire telles que les méthodes basées sur les points, [57, 43, 71], celles basées sur des grilles [34, 13, 7] ou bien les approches de Monte Carlo [68], utilisent des calculs approchés.

Imprécision des Paramètres et Vision Artificielle

Considérons maintenant des robots utilisant la perception visuelle, et dont les observations proviennent d'algorithmes de vision basés sur de l'apprentissage statistique. (*cf.* figure 3). Dans cette situation, le robot utilise un *classifieur* pour reconnaître les objets dans les images : le classifieur est censé renvoyer le nom de l'objet qui se trouve dans l'image, et fait quelquefois des erreurs avec une faible probabilité. (*cf.* matrice de confusion de la figure 5).

Le classifieur est calculé en utilisant une *base de données d'entraînement* (comme *NORB*, *cf.* figure 4, mis en ligne par les auteurs à l'adresse `http://www.cs.nyu.edu/~ylclab/data/norb-v1.0/`).

Les figures (4) et (5) illustrent l'exemple d'un classifieur calculé pour une mission dronique dans laquelle les caractéristiques d'intérêt (les état du système) sont liées à la présence (ou l'abscence) d'animaux (*animals*), voitures (*cars*), humains (*humans*), avions (*planes*) ou camions (*trucks*) : le problème statistique du calcul d'un classifieur permettant de reconnaitre de tels objets dans les images est appelé *classification multi-classes*.

Puisque le classifieur est appris à partir d'une base de données d'images, son comportement, et donc ses performances, (*i.e.* sa fréquence de bonne prédiction) est inévitablement dépendant de la base de données. Cela pose un pose un problème si la variabilité de la base de donnée est trop faible : dans ce cas, le comportement probabiliste du classifieur sera dépendant de ces images en particulier et le système robotique aura des mauvaises capacités d'observation lorsque la mission implique des images trop différentes de celles présentes dans la base de données.

Certaines bases de données à grande variabilité existent (par exemple *NORB*, figure 4, bien que la variabilité pourrait être idéalement supérieure) : notons cependant qu'avec ces bases de données, la performance de vision est réduite, ou bien, au moins, de bonne performances sont difficilement atteignables

base de données d'images *NORB* : $(\text{image}_i, \text{étiquette}_i)_{i=1}^{N}$

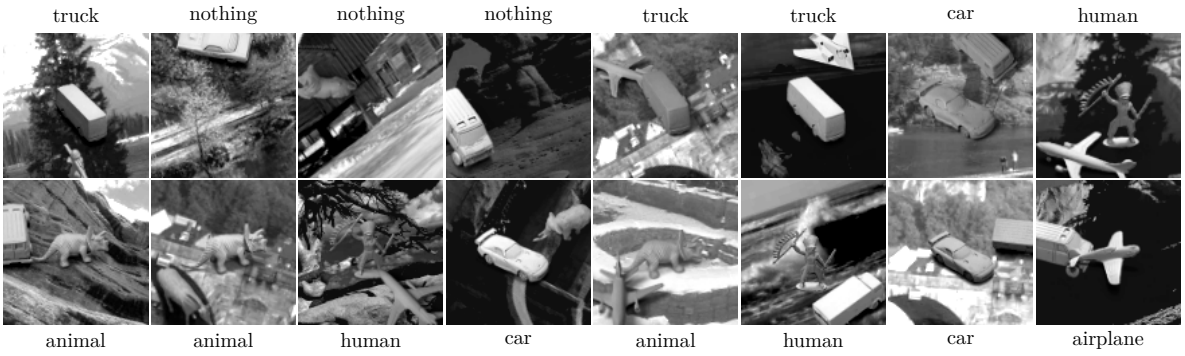


FIGURE 4 – *Exemple de base de données pour la vision artificielle : la base de données d'images étiquetées NORB [45], destinée à l'apprentissage statistique. La taille de NORB est supérieure à* 3.10^5*, et les images de cette base de données représentent des objets de ces* 5 *classes : "animal", "car", "human", "nothing", "plane" et "truck". Chaque élément d'une base de données d'images est composé d'une image (par exemple une image montrant une voiture) et une étiquette correspondant à la classe de l'objet représenté par l'image (dans l'exemple précédent, l'étiquette est "car"). Cette base de données peut être utilisée afin de calculer un classifieur par apprentissage supervisé. Dans le but de discerner les positions des cibles, une image est étiquetée avec le nom de l'objet qui est au centre ("nothing" si il n'y a rien au centre de l'image).*

Une matrice de confusion peut être calculée (*cf.* figure 5) en utilisant une telle base de données étiquetée. Elle doit être différente de celle utilisée pour l'entraînement et est appelée *base de données de test* : la fréquence des observations peut être déduit de cette matrice, en normalisant les lignes ce qui en fait des distributions de probabilité. Une ligne correspond à un objet de la scène et les probabilités de cette ligne sont les probabilités d'observation, *i.e.* chaque valeur de probabilité est la fréquence avec laquelle le classifieur renvoie le nom de l'objet de la colonne correspondante. Ici le classifieur a été calculé à partir d'un Réseau Convolutionnel [46].

Ces probabilités peuvent être utilisées pour définir la fonction d'observation $\mathbf{p}(o' \mid s', a)$ introduite au-dessus. Cette approche soulève encore le problème de la représentativité de la base de données pour la mission voulue. Si la base de donnée de test n'est pas représentative, ces probabilités d'observation risquent de ne pas être fiables, et le PDMPO mal défini : cependant, comme montré par l'équation (2) la mise à jour de l'état de croyance nécessite la connaissance parfaite de la fonction d'observation.

Finalement, si les bases de données considérées sont étiquetées plus précisément, (comme *NORB*, qui inclut des informations telles que la luminosité, ou l'échelle de l'objet), nous pouvons imaginer que les probabilités d'observation calculées (à partir de la matrice de confusion) seraient plus fiables, ou la performance de vision améliorée (puisque la séparation demandée au classifieur est plus simple avec cette précision). Cependant, comme cela implique l'utilisation de plus d'observations ou d'états, le PDMPO est plus dur à résoudre.

En conclusion, l'utilisation du modèle PDMPO fait l'hypothèse que les distributions de probabilité régissant le problème doivent être toutes connues parfaitement : malheureusement ces fréquences ne sont pas connue précisément en pratique. L'imprécision à propos de ces probabilités doit être prise en compte pour rendre le robot autonome en toutes circonstances. En général, le calcul des distributions de probabilité d'un PDMPO nécessite assez de tests pour chaque paire état-action, ce qui est dur à effectuer en pratique.

Quelques variations du cadre PDMPO a été constuit dans le but de prendre en compte l'imprécision sur les distributions de probabilité du modèle, aussi appelé *l'imprécision des paramètres.*

Travaux Tenant Compte de l'Imprécision des Paramètres

Ici, les fonctions de transition et d'observation, *i.e.* $\mathbf{p}(s' \mid s, a)$ et $\mathbf{p}(o' \mid s', a)$, $\forall (s, s', o', a) \in \mathcal{S}^2 \times \mathcal{O} \times \mathcal{A}$, sont appelées *paramètres* du PDMPO, ou encore *paramètres* du modèle. A notre connaissance, le premier modèle construit dans le but de gérer l'imprécision des paramètres est nommé PDMPOPI, pour *PDMPO à Paramètres Imprécis* [37]. Dans ce travail, chacun des paramètres du PDMPO est remplacé par un ensemble de paramètres possibles. Dans ce travail, une *croyance du second ordre* est introduite : elle est définie comme étant la distribution de probabilité sur les paramètres du modèles.

Un autre travail, appelé *PDMPO à Paramètres Bornés* (PDMPOPB) [50], traite aussi de l'imprécision des paramètres : dans ce travail, l'imprécision sur chaque paramètre est décrit à l'aide d'une borne supérieure et inférieure sur les distributions possibles. Aucune croyance du second ordre n'est introduite ici. Cependant, résoudre un PDMPOPB est similaire, dans l'esprit, à la résolution des PDMPOPI [37] : la flexibilité amenée par l'imprécision des paramètres est utilisée pour rendre les calculs les plus faciles possible, et le critère utilisé n'est pas explicite. Le problème majeur de ces approches (PDMPOPI et PDMPOPB) est que l'imprécision des paramètres n'est pas géré dans un but de robustesse, comme une approche pessimiste (pire cas), mais dans un but de simplification.

Un travail plus récent traite le problème de manière pessimiste et est donc appelé *PDMPO robuste* [53]. Inspiré par le travail correspondant dans le cas complètement observable (appelé PDM incertain [51]), ce travail utilise le critère dit *maximin*, ou du *pire cas*, qui vient de la théorie des jeux : dans ce cadre, une stratégie optimale maximise le plus petit critère parmi ceux induits par chaque paramètre possible. Si l'imprécision des paramètres n'est pas stationnaire, *i.e.* peut changer à chaque étape de temps, la stratégie optimale associée (au sens du maximin) peut être calculée en utilisant la *Programmation Dynamique* [4]. Cependant, lorsque l'imprécision des paramètres est stationnaire, les choses se compliquent : les calculs proposés mènent à une approximation de la stratégie optimale, puisque le critère utilisé est une borne inférieure du critère désiré. Pourtant, une hypothèse stationnaire pour l'imprécision des paramètres semble mieux adaptée lorsque le PDMPO est stationnaire.

Bien que l'utilisation d'ensembles de distributions de probabilité rend le modèle plus adapté à la réalité du problème (les paramètres sont imprécis en pratique), les prendre en compte augmente violemment la complexité du calcul d'une politique optimale (par exemple, lors de l'utilisation du critère maximin). Comme expliqué précédemment, résoudre un PDMPO est déjà une tâche très ardue, donc l'utilisation d'un cadre menant à des calculs plus complexes ne semble pas être une approche satisfaisante. En effet, modéliser le problème de manière

animal	human	plane	truck	car	nothing		
3688	575	256	48	144	149	animal	75.885%
97	4180	81	20	225	257	human	86.008%
292	136	3906	237	202	87	plane	80.370%
95	1	44	4073	514	133	truck	83.807%
129	3	130	1283	3283	32	car	67.551%
154	283	36	63	61	4263	nothing	87.716%

FIGURE 5 – *Exemple d'une matrice de confusion en classification multi-classes : cette matrice est calculée avec une base de données de test, différente de la base de données d'apprentissage. Chaque ligne ne considère que les images d'un certain objet, et les nombres représentent les réponses du classifieur : par exemple,* 3688 *images d'animaux sont bien reconnues, mais* 575 *sont confondues avec un humain. La proportion de réponses correctes pour cet objet est* 80.223%*. L'environnement Torch7 [18] a été utilisé pour obtenir un classifieur, et pour calculer cette matrice à partir du classifieur et de la base de donnée de test.*

trop fine mène à l'utilisation de nombreuses approximations dans les calculs en pratique, sans réel contrôle ou estimation de ces approximations comme dans le cas des PDMPOPI et des PDMPOPB. Il est peut-être plus judicieux de commencer avec un modèle plus simple qui peut être résolu plus facilement en pratique.

Un autre problème pratique du modèle PDMPO peut être mentionné : il concerne la définition de l'état de croyance durant les premières étapes du processus, et plus généralement, la manière avec laquelle la connaissance de l'agent est représentée.

Modéliser l'Ignorance de l'Agent

L'état de croyance initial b_0, ou distribution de probabilité *a priori* sur l'état du système, prend part dans la définition du PDMPO. Étant donné un état du système $s \in \mathcal{S}$, $b_0(s)$ est la fréquence de l'évènement "l'état initial est s". Cette quantité peut être dure à calculer rigoureusement, surtout lorsque le nombre d'expériences passées est limité : cette raison a déjà été invoquée au-dessus, menant à l'imprécision des fonctions de transition et d'observation.

Considérons l'exemple d'un robot qui est pour la première fois dans une salle dont la position de la sortie est inconnue (état de croyance initial) et qui doit trouver la sortie et l'atteindre. En pratique, aucune expérience ne peut être répétée dans le but d'extraire une fréquence de position pour cette sortie. Dans ce genre de situation, l'incertitude n'est pas due à un événement aléatoire, mais à un manque de connaissance : aucun état de croyance initial fréquentiste ne peut être utilisé pour définir le modèle.

L'agent peut aussi croire fortement que la sortie est positionnée dans un mur, comme dans la plupart des salles, mais il attribue quand-même une très petite probabilité p_ϵ au fait que la sortie peut être un escalier au plein milieu de la salle. Même s'il est très peu probable que ce soit le cas, cette seconde option doit être prise en compte dans l'état de croyance, sinon la règle de Bayes (*cf.* équation 2) ne peut pas le mettre à jour correctement si la sortie est vraiment au centre de la pièce. Quantifier p_ϵ sans expérience passée n'est pas une chose facile du tout, et ne se repose sur aucune justification rationnelle, mais peut impacter fortement la stratégie de l'agent.

L'état initial du système peut être délibérément défini comme étant inconnu, avec strictement aucune information probabiliste : considérons une mission robotique pour laquelle une partie de l'état du système, décrivant un fait que le robot est censé découvrir par lui-même, est initialement complètement inconnu. Dans un contexte d'exploration robotique, la position ou la nature d'une cible, ou encore la position initiale du robot, peut être défini comme étant absent de la connaissance de l'agent. Les approches classiques initialisent l'état de croyance par une distribution de probabilité uniforme. (*i.e.* sur toutes les positions possibles du robot/cible, ou sur toutes les natures possibles de cible), mais cette réponse provient de l'interprétation subjective des probabilités [20, 31]. En effet, les probabilités sont les mêmes puisqu'aucun événement n'est plus plausible qu'un autre : cela correspond à des paris égaux. Cependant, les mises à jour de l'état de croyance (*cf.* équation 2) mène fatalement à un mélange de probabilités fréquentistes, *i.e.* les fonctions de transition et observation, avec cet état de croyance initial qui est une distribution de probabilité subjective : cela n'a pas toujours de sens, et dans notre cas cette approche est douteuse. Ainsi, l'utilisation des PDMPO dans ces contextes fait face à la difficulté de représenter l'ignorance de l'agent.

Problème Général

Les sections précédentes ont présenté certain problèmes rencontrés en pratique lors de l'utilisation des PDMPO pour calculer des stratégies, en particulier dans le cadre robotique. La très grande complexité du calcul d'une stratégie optimale est un premier problème : les

missions robotiques sont souvent des problèmes à grandes dimensions, dont le calcul d'une stratégie suffisamment proche d'une stratégie optimale est impossible, du fait d'un trop grand temps de calcul ou d'un manque de mémoire vive. Ensuite, il a été mis en évidence que les distributions de probabilité définissant le PDMPO ne sont pas toujours connues précisément : par exemple, la fonction d'observation peut être difficile à définir lorsque les observations proviennent d'un algorithme de vision artificielle complexe. Enfin, le problème de la gestion de la connaissance et de l'ignorance de l'agent a été discuté : il n'y a pas de réponse formelle concernant la manière de représenter le manque de connaissance initial de l'agent à propos du monde dans lequel il évolue. La difficulté vient du fait que les PDMPO classiques (probabilistes) autorisent uniquement l'usage de distribution de probabilité fréquentiste, alors qu'un autre outil mathématique semble nécessaire.

Ces problèmes forment le point de départ de notre travail. En effet, ce dernier consiste à contribuer au problème du calcul d'une stratégie pour des domaines partiellement observables. Les stratégies calculées doivent permettre au robot de remplir sa mission aussi bien que possible, dès la première exécution *i.e.* le calcul de stratégie est opéré avant toute réelle exécution de la mission.

Le chalenge général guidant ce travail est de procéder aux calculs de stratégies en utilisant seulement les données et les connaissances vraiment disponibles en pratique, au lieu d'utiliser les PDMPO classiques, aux calculs très complexes et aux paramètres difficiles à définir. En d'autre mots, cela revient à prêter attention aux problèmes soulevés au-dessus : la complexité de calcul, l'imprécision du modèle, et la gestion de la méconnaissance de l'agent. Comme expliqué par la suite, la théorie des possibilités qualitatives [28] semble pouvoir répondre aux problèmes soulevés.

Une Théorie Qualitative

Cette théorie est généralement introduite en définissant une échelle qualitative $\mathcal{L}$, qui peut être définie par $\left\{0, \frac{1}{k}, \frac{2}{k}, \ldots, 1\right\}$, avec $k > 1$, ou tout autre ensemble fini totalement ordonné. Les valeurs de cette échelle ne sont pas importantes car elle servent seulement à matérialiser un ordre : nous utilisons donc le terme de *degré de possibilité* afin d'expliciter cette remarque. Le plus petit élément de l'échelle qualitative est noté 0, et le plus grand, 1. La section suivante clarifie pourquoi l'utilisation de ce cadre qualitatif est bénéfique en matière de complexité et de modélisation.

Notons tout d'abord les similarités entre la théorie des probabilités et des possibilités : une distribution de possibilité sur $\mathcal{S}$ est une fonction $\pi : \mathcal{S} \to \mathcal{L}$ telle que $\max_{s \in \mathcal{S}} \pi(s) = 1$. L'opérateur de marginalisation est l'opérateur maximum (max) : étant donné une distribution de possibilité jointe $\pi : \mathcal{S} \times \mathcal{O} \to \mathcal{L}$, la distribution de possibilité marginale sur $\mathcal{S}$ est $\pi(s) = \max_{o \in \mathcal{O}} \pi(s, o)$. De plus, l'opérateur minimum (min) est utilisé pour calculer une distribution de possibilité jointe $\pi(s, o)$, $\forall (s, o) \in \mathcal{S} \times \mathcal{O}$ à partir d'une distribution marginale $\pi(s)$, $\forall s \in \mathcal{S}$ et d'une distribution conditionnelle $\pi(o \mid s)$, $\forall (o, s) \in \mathcal{S} \times \mathcal{O} : \pi(s, o) = \min\{\pi(s), \pi(o \mid s)\}$. Ainsi, il est possible d'apprivoiser la théorie des possibilités en remplaçant l'opérateur + de la théorie des probabilités par l'opérateur max, et l'opérateur $\times$ par min.

PDMPO Qualitatifs Possibilistes

Une alternative possibiliste qualitative du modèle PDMPO a été proposée dans [62] : ce modèle s'appelle PDMPO qualitatif possibiliste, et est noté π-PDMPO. Un π-PDMPO est simplement un PDMPO avec des distributions possibilistes qualitatives comme paramètres, au lieu de distributions de probabilité. Comme les π-PDMPO sont qualitatifs, l'homogue de la fonction de récompense, appelée *fonction de préférence*, est aussi qualitative : en effet, la

fonction de préférence est à valeurs dans l'échelle possibiliste qualitative finie $\mathcal{L}$, et donc n'est pas additive.

L'une des propriétés les plus intéressantes des π-PDMPO est la simplification du calcul de la stratégie. En effet, les algorithmes proposés pour résoudre les PDMPO probabilistes sont souvent basés sur l'ensemble des états de croyance appelé *espace des croyances*. L'espace des croyances est infini dans le cas général : chaque étape de temps mène à un nombre fini d'états de croyance suivants, donc cet espace est dénombrable. Dans le but d'obtenir des propriétés utiles et des moyens de calculer une stratégie, l'ensemble de toutes les distributions de probabilité sur l'espace d'état $\mathcal{S}$ est souvent considéré, *i.e.* le simplex continu $\mathbb{P}^{\mathcal{S}} = \{\, \mathbf{p} : \mathcal{S} \to [0,1] \mid \sum_{s\in\mathcal{S}} \mathbf{p}(s) = 1, \text{ and } \mathbf{p}(s) \geqslant 0, \forall s \in \mathcal{S} \,\}$. La taille infinie de l'espace des croyances explique en partie pourquoi les PDMPO probabilistes sont vraiment difficiles à résoudre. Au contraire, les π-PDMPO ont un espace de croyances fini. En effet, le nombre de distributions de possibilité qualitatives sur les états du système $\mathcal{S}$ est inférieur à $\#\mathcal{L}^{\#\mathcal{S}}$, puisque l'échelle qualitative $\mathcal{L}$ est finie. La version complètement observable d'un π-PDMPO est appelé π-PDM : comme expliqué dans la section I.4 du chapitre I, tout π-PDMPO se réduit à un π-PDM dont l'espace d'état est l'ensemble des états de croyance possibilistes qualitatifs, et dont la taille est exponentielle en fonction du nombre d'états du système. Dans les travaux [32, 33, 62], il est montré que la complexité d'un π-PDM est plus faible que la complexité d'un PDM probabiliste, qui est polynomiale [54] : la complexité d'un π-PDMPO est au pire exponentiel en la description du problème, tandis qu'un PDMPO probabiliste peut être indécidable [47].

En plus de la simplification des calculs, le π-PDMPO peut être vraiment intéressant pour nos problèmes de modélisation. En effet, dans le cas d'un robot utilisant un algorithme de vision artificielle (*cf.* figure 3), nous avons précédemment mis en évidence la difficulté à définir rigoureusement les fonctions d'observation probabilistes : les probabilités des réponses des algorithmes de vision dans le contexte d'une mission robotique sont mal connues et difficiles à définir en pratique. Il est plus facile de trouver des estimations qualitatives des performances de reconnaissance de l'algorithme : le modèle π-PDMPO ne requiert que des données qualitatives, donc il permet de construire un modèle sans l'utilisation d'informations autres que de celles vraiment disponibles. Par exemple, la matrice de confusion de la figure 5 peut mener à une fonction d'observation qualitative possibiliste tenant compte uniquement de la manière dont les fréquences de réponses sont classées : en présence d'un humain (*i.e.* deuxième ligne), la réponse la plus fréquente est "*human*", la deuxième réponse est "*nothing*", la troisième est "*car*", etc. Donc, la distribution de possibilité correspondante est telle que, conditionnellement à la présence d'un humain, le degré de possibilité de la réponse "*human*" est plus grand que le degré de possibilité de la réponse "*nothing*", qui est plus grand que le degré de possibilité de la réponse "*car*", etc. Au lieu d'attribuer des fréquences qui ne sont pas vraiment fiable en pratique, le modèle possibiliste qualitatif exprime naturellement ces imprécisions relatives au problème.

Enfin, notons que la distribution de possibilité constante, dont les degrés sont tous égaux à 1 (élément maximal de $\mathcal{L}$), représente l'ignorance totale : cette distribution peut être utilisée pour définir l'état de croyance initial lorsqu'il doit représenter un agent qui ignore initialement une situation donnée. Donc, le modèle π-PDMPO permet une modélisation formelle du manque de connaissance de l'agent.

L'utilisation de la théorie des possibilités qualitatives [29] est donc étudié dans ce travail, puisqu'il semble être capable d'à la fois simplifier un PDMPO, et de modéliser l'imprécision des paramètres et l'ignorance associées aux missions robotiques. Ce cadre, en effet, simplifie les calculs, est capable de représenter le problème avec seulement les données disponibles, et modélise le manque de connaissance : ainsi cette théorie offre des solutions aux trois problèmes

mis en évidence précédemment. Cependant, notons qu'un cadre qualitatif ne permet pas de manipuler de l'information fréquentiste.

A notre connaissance, une étude plutôt limitée du modèle π-PDMPO existe dans la littérature jusqu'à aujourd'hui : en fait, le travail [62] semble être le seul, proposant à la fois une définition des π-PDMPO et un exemple jouet pour illustrer ce modèle. La version complètement observable (π-PDM) a généré plus de travaux [64, 63, 77].

Description de l'Étude

Cette thèse contribue à déterminer dans quelle mesure la thérie de possibilités qualitatives peut contribuer à la *planification dans l'incertain dans des domaines partiellement observables*, et plus généralement à la gestion séquentielle de l'incertitude, en matière de simplification des calculs et de modélisation. Elle présente de récentes contributions dans l'utilisation de cette théorie pour la planification dans l'incertain et la représentation des connaissances, avec une utilisation quasi systématique des modèles graphiques [40, 6, 10].

La fin de cette introduction décrit la structure de cette thèse. En effet, chacune des sections suivantes correspond à un chapitre de notre travail, et détaille son contenu.

État de l'Art

Les PDMPO qualitatifs et possibilistes constituent l'objet central de cette thèse : le *premier chapitre* est consacré à ce modèle. Une présentation rapide de la théorie des possibilités qualitatives est suivie de la présentation du modèle entièrement observable (π-PDM), puis du modèle partiellement observable (π-PDMPO). Comme noté précédemment, à notre connaissance, seul un article de dix pages a déjà traité des π-PDMPOs.

Mises à jour naturelles du modèle possibiliste qualitatif

Le *deuxième chapitre* propose quelques extensions au travail [62]. Tout d'abord, est construite une version possibiliste qualitative des PDM à observabilité mixte [52, 2] dans laquelle quelques variables d'état sont complètement observables. Elle est appelée π-PDMOM et généralise à la fois les π-PDM et les π-PDMPO. Cette contribution réduit considérablement la complexité de résolution des π-PDMPO, en manipulant de manière plus fine l'information des environnements dont certaines variables sont complètement observables. Par exemple, le niveau de batterie d'un robot peut être considéré comme une information directement observable pour la prise de décision, menant à des calculs plus simples. Plus généralement, il est très courant de manipuler des variables visibles en robotique [52].

Ensuite, un critère qualitatif est proposé pour pouvoir traiter des missions dont la durée n'est pas connue à l'avance : l'algorithme faisant le calcul de la stratégie optimale associée est alors présenté. Cet algorithme est utilisé pour calculer une stratégie pour une mission de reconnaissance de cible : les résultats expérimentaux comparent les exécutions utilisant cette stratégie à celles utilisant la stratégie d'un algorithme pour PDMPO probabiliste, dans des situations où la dynamique probabiliste des observations est en fait mal définie.

Notons que ces résultats expérimentaux constituent, à notre connaissance, la première utilisation du cadre π-PDMPO. Ils mettent aussi en évidence un comportement intéressant de l'état de croyance possibiliste qualitatif dans certaines situations détaillées ensuite. Les principales contributions de ce chapitre ont été publiées dans [24]. Il est à souligner que les résultats expérimentaux n'auraient pas pu être effectués sans les deux premières contributions, qui permettent de ne pas fixer un horizon arbitraire et d'alléger les calculs.

Cependant, ces contributions ne sont pas suffisantes pour atteindre un temps de calcul compétitif, ou pour pouvoir traiter des problèmes robotiques réels : l'orientation du second chapitre est dictée par cette remarque.

Modèles Factorisés et Algorithmes Symboliques

Les problèmes robotiques traités dans le chapitre précédent sont assez petits pour permettre à l'algorithme proposé de calculer une stratégie en un temps raisonnable. Les contributions du *troisième chapitre* permettent la résolution de problèmes de planification structurés dont l'espace des états est plus grand.

La première partie de ce chapitre propose d'introduire les *π-PDMOM* factorisés : il sont définis par des hypothèses d'indépendance supplémentaires. Les grands problèmes de planification satisfaisant ces hypothèses peuvent être résolus plus facilement : inspiré par l'algorithme pour PDM probabilistes, *SPUDD* [36], nous avons conçu un algorithme nommé *PPUDD* pour résoudre les π-PDMOM factorisés en utilisant des *arbres de décision algébriques* (*ADDs*). L'intuition motivant cette contribution, est que le calcul entre *ADDs* est moins coûteux en temps et en mémoire dans le cadre qualitatif possibiliste que dans le cadre probabiliste : les opérations qualitatives devraient mener à des *ADDs* plus petits puisque la somme et le produit sont remplacés par les opérateurs min et max. Ces derniers produisent des *ADDs* avec potentiellement moins de feuilles, car ils ne créent pas de nouvelles valeurs.

Les hypothèses d'indépendance définissant le modèle π-PDMOM factorisé concernent les variables représentant les états de croyance successifs. De plus, les variables définissant un π-PDMOM sont celles représentant les états du système et les observations. C'est pourquoi la section de ce chapitre qui suit propose des conditions suffisantes sur les variables d'état et d'observation menant aux indépendances désirées entre les variables de croyances. Un exemple robotique est utilisé en guise d'illustration de ces conditions. Puisque les preuves utilisent le concept graphique appelé *d-Séparation* [74], ces conditions mènent aussi à l'indépendance des variables de croyance dans le cadre des PDMOM probabilistes.

Les performances de notre algorithme *PPUDD* sont ensuite comparées à celles des homologues probabilistes, en termes de temps de calcul et avec des critères mesurant la réussite de la mission. Enfin, la dernière partie de ce chapitre décrit les résultats de *PPUDD* lors de la compétition internationale de planification probabiliste [1]. Nous avons participé à la compétition dans le but de tester les performances de *PPUDD* contre celles des algorithmes probabilistes, en termes d'espérances de la somme des récompenses, et sur de nombreux problèmes de planification.

Certaines contributions de ce chapitre ont été publiées dans [25]. Les nombreux problèmes de planification de la compétition mettent aussi en évidence certaines failles de notre algorithme, lorsqu'il est utilisé pour approximer le calcul d'une stratégie pour un problème probabiliste dans le but de bénéficier de calculs qualitatifs qui sont plus simples. De plus, bien que notre algorithme est meilleur que certains algorithmes utilisant des *ADDs* (notamment sont homologue direct, *SPUDD*), les algorithmes de la compétition utilisant des recherches dans l'espace d'état [38, 42] obtiennent de meilleurs résultats. L'approche proposée dans [23] prend en compte ces problèmes.

1. `https://cs.uwaterloo.ca/~mgrzes/IPPC_2014/`

État de l'Art

I

Le principal sujet de cette thèse est le processus décisionnel markovien partiellement observable (PDMPO), ou plus précisément son homologue possibiliste qualitatif. L'utilisation pratique du modèle probabiliste a été critiquée en introduction, cependant ce modèle résume de manière appropriée les principales caractéristiques d'un système robotique : son homologue possibiliste étant très similaire, son étude semble prometteuse. Tout d'abord, la théorie des possibilités qualitative est présentée dans le but d'introduire les *processus décisionnels markoviens possibilistes qualitatifs* (π-PDM) et enfin les *processus décisionnels markoviens partiellement observables* (π-PDMPO) qui constituent le point de départ de notre travail.

I.1 Théorie des Possibilités Qualitatives

Les *ensembles flous* ont été introduits par Lotfi Zadeh [79], et étudiés par Didier Dubois [19] et Henri Prade : leur contributions ont mené à la fondation de la théorie des possibilités [27].

Comme la théorie des probabilités, cette théorie est basée sur une mesure d'incertitude, appelée *mesure de possibilité*. Contrairement à la mesure de probabilité $\mathbb{P}$ qui est une mesure classique, la mesure de possibilité, notée Π, est une *mesure floue*, ou *capacité*. Pour faire simple, une mesure floue n'est pas supposée additive, mais juste *monotone*, *i.e.* si $A \subset B$ alors la mesure de A est plus petite que la mesure de B. Dans cette thèse, les mesures de possibilité vont concerner uniquement des ensembles finis comme l'ensemble des états $\mathcal{S}$ et l'ensemble des observations $\mathcal{O}$.

Formellement, une mesure de possibilité est définie comme suit :

Définition I.1.1 (*Mesure de Possibilité*)

Une mesure de possibilité sur l'ensemble fini Ω est une fonction $2^\Omega \to [0,1]$ telle que
- $\Pi(\Omega) = 1$ *(normalisation) ;*
- $\forall A, B \subset \Omega$, $\Pi\{A \cup B\} = \max\{\Pi(A), \Pi(B)\}$ *(maxitivité).*

La théorie des probabilité modélise l'incertitude due à la variabilité des événements : en pratique, les probabilités sont les fréquences estimées des événements, définis comme le vrai modèle de variabilité des événements. Une autre interprétation de cette théorie est celle des probabilités subjectives définies par De Finetti [20] : la valeur de probabilité d'un événement est un pari échangeable, relatif aux connaissances d'une personne donnée.

La théorie des possibilités est dédiée à l'incertitude due à un manque de connaissance, ou une imprécision à propos d'un événement. La théorie des possibilités quantitatives est un cas particulier de probabilités imprécises *i.e.* une mesure de possibilité quantitative Π représente un ensemble particulier de mesures de probabilité définies sur Ω.

Contrairement à la théorie des possibilités quantitatives, la théorie des possibilités qualitatives utilise des mesures de possibilité dont les valeurs sont définies sur n'importe quelle

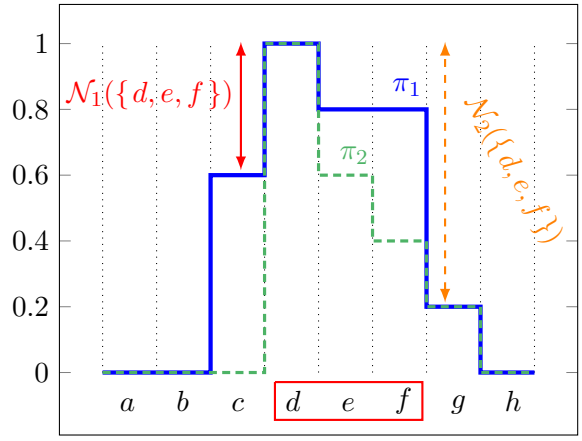


FIGURE I.1 – *Exemple de deux distributions de possibilité sur* $\Omega = \{a, b, c, d, e, f, g, h\}$ *:* π_1 *(ligne bleue continue) et* π_2 *(ligne verte pointillée), avec* π_2 *qui est plus spécifique que* π_1*. La mesure de nécessité* $\mathcal{N}_1$ *associée à* π_1 *est évaluée sur l'événement* $\{d, e, f\} \subset \Omega$ *: le degré de nécessité est égal à* $0.4 = 1 - 0.6$*, comme illustré par les flèches rouges continues. La mesure de nécessité* $\mathcal{N}_2$ *associée à* π_2 *est évaluée sur le même événement : le degré de nécessité est égal à* $0.8 = 1 - 0.2$*, comme illustré par les flèches oranges en pointillées.*

échelle ordonnée. Cette théorie nous permet de raisonner, même lors d'un manque d'informations quantitatives : la seule information donnée par une mesure de possibilité qualitative est l'ordre de plausibilité entre les événements *i.e.* pour $A, B \subseteq \Omega$, l'information "l'événement A est moins plausible que l'événement B", qui s'écrit $\Pi(A) \leqslant \Pi(B)$. Ainsi les mesures de possibilité qualitatives Π sont souvent définies comme des fonctions $2^\Omega \to \mathcal{L}$, où $\mathcal{L}$ est un ensemble fini, appelé *échelle possibiliste*, et possédant un ordre total. Dans ce travail, l'échelle possibiliste est définie par $\mathcal{L} = \left\{0, \frac{1}{k}, \ldots, 1\right\}$ pour simplifier les notations.

Nous pouvons définir la *distribution de possibilité* π comme suit : $\forall \omega \in \Omega$, $\pi(\omega) = \Pi(\{\omega\})$. D'après la définition I.1.1, une mesure de possibilité Π est entièrement définie par la distribution associée π. Pour chaque distribution (ou mesure) de possibilité, une mesure duale, appelée mesure de nécessité, peut être définie : le degré de nécessité d'un événement augmente si le degré de possibilité de l'événement contraire décroît.

Définition I.1.2 (*Mesure de nécessité associée à* Π)

La mesure de nécessité associée à la mesure de possibilité Π *est la mesure floue* $\mathcal{N} : 2^\Omega \to \mathcal{L}$ *telle que* $\forall A \subset \Omega$*,*

$$\mathcal{N}(A) = 1 - \Pi(\overline{A}),$$

où $\overline{A}$ *est l'événement complémentaire de* A *dans* Ω*.*

L'ignorance totale est modélisée par une distribution de possibilité π telle que $\forall \omega \in \Omega$, $\pi(\omega) = 1$ *i.e.* n'importe quel événement élémentaire est possible. Dans ce cas, $\forall A \subseteq \Omega$, $A \neq \Omega$, $\mathcal{N}(A) = 1 - \Pi(\overline{A}) = 1 - \max_{\omega \in \overline{A}} \Pi(\omega) = 0$: aucun événement n'est nécessaire, sauf l'univers entier Ω.

La connaissance parfaite que l'événement élémentaire $\omega_A \in \Omega$ est elle modélisée par une distribution de possibilité π telle que $\pi(\omega_A) = 1$ et $\pi(\omega) = 0$, $\forall \omega \neq \omega_A$. Le degré de nécessité du singleton $\{\omega_A\}$ est aussi égal à 1 : $\mathcal{N}(\{\omega_A\}) = 1 - \Pi(\overline{\{\omega_A\}}) = 1$. L'événement élémentaire ω_A est nécessaire, et tous les autres événements élémentaires ont un degré de nécessité nul : si $\omega_B \neq \omega_A$, $\mathcal{N}(\{\omega_B\}) = 1 - \Pi(\overline{\{\omega_B\}}) = 1 - \pi(\omega_A) = 0$.

Généralement, une distribution de possibilité est plus informative, ou plus *spécifique*, que l'ignorance totale, et moins *spécifique* que la connaissance parfaite :

Définition I.1.3 (*Spécificité*)

Une distribution de possibilité π_2 est plus spécifique que la distribution de possibilité π_1, si $\forall \omega \in \Omega$,

$$\pi_2(\omega) \leqslant \pi_1(\omega).$$

Les notions de nécessité et de spécificité sont illustrées figure I.1.

Les concepts de la théorie des possibilités qualitatives nécessaires à la compréhension de notre travail ont été présentés. Nous présentons maintenant l'homologue du critère basé sur l'espérance dans les modèles probabilistes : les critères qualitatifs utilisés dans les modèles possibilistes qualitatifs.

I.2 Critères Qualitatifs

L'espérance utilisée comme critère dans les PDMPO probabilistes, est simplement l'intégrale de la la fonction de récompense contre la mesure de probabilité. Le concept d'intégrale a été étendue aux mesure floues : quand la mesure est quantitative, l'extension est appelée *intégrale de Choquet* [17]. Dans le cas des mesures qualitatives, l'objet résultant est l'*intégrale de Sugeno* [72]. Nous définissons donc ici l'intégrale de Sugeno :

Définition I.2.1 (*Intégrale de Sugeno*)

L'intégrale de Sugeno d'une fonction $f : \Omega \to \mathcal{L}$ contre la capacité (mesure floue) $\mu : 2^\Omega \to \mathcal{L}$ est

$$\mathbb{S}_\mu[f] = \max_{i=1}^{\#\Omega} \min\{f(\omega_i), \mu(A_i)\} \tag{I.1}$$

$$= \min_{i=1}^{\#\Omega} \max\{f(\omega_i), \mu(A_{i+1})\} \tag{I.2}$$

où $f(\omega_1) \leqslant \ldots \leqslant f(\omega_{\#\Omega})$, $A_i = \{\omega_i, \omega_{i+1}, \ldots, \omega_{\#\Omega}\}$ et $A_{\#\Omega+1} = \emptyset$.

Comme illustré dans la figure I.2, l'intégrale de Sugeno de $f : \Omega \to \mathcal{L}$ contre la mesure floue μ est le plus grand degré $\lambda \in \mathcal{L}$ tel que la mesure μ de $\{\omega \mid f(\omega) \geqslant \lambda\}$ est plus grande ou égale à λ. Par exemple, le h-indice (ou indice de Hirsh) est l'intégrale de Sugeno de la fonction *papier* $\mapsto$ *#citations* contre la mesure de comptage.

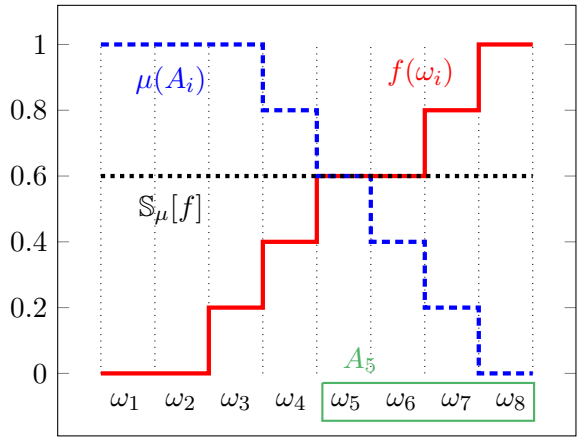


Figure I.2 – *Illustration du résultat de l'intégrale de Sugeno : l'axe des abscisses représente l'ensemble $\Omega = \{\omega_1, \ldots, \omega_{\#\Omega}\}$, où $\forall i \in \{1, \ldots, \#\Omega - 1\}$, $f(\omega_i) \leqslant f(\omega_{i+1})$. L'axe des ordonnées est $\mathcal{L}$. La courbe rouge représente les degrés $f(\omega_i)$, la bleue en pointillées représente les degrés $\mu(A_i)$ avec $A_i = \{\omega_i, \ldots, \omega_{\#\Omega}\}$, et la noire est le résultat de l'intégrale de Sugeno.*

L'intégrale de Sugeno contre une mesure de possibilité et celle contre la nécessité, mènent à deux critères pour la planification. Ces intégrales se réécrivent comme suit :

Théoreme 1 (*L'intégrale de Sugeno contre les mesures de possibilité et de nécessité*)

$$\mathbb{S}_{\Pi}[f] = \max_{i=1}^{\#\Omega} \min\{ f(\omega_i), \pi(\omega_i) \}, \tag{I.3}$$

$$\mathbb{S}_{\mathcal{N}}[f] = \min_{i=1}^{\#\Omega} \max\{ f(\omega_i), 1 - \pi(\omega_i) \}. \tag{I.4}$$

sont les réécritures des intégrales de Sugeno contre les mesures de possibilité et de nécessité.

Les critères possibilistes qualitatifs, *i.e.* les fonctions $\mathcal{A} \to \mathcal{L}$ mesurant la validité des actions étant donné un modèle possibiliste et une fonction de préférence, a été proposé dans [66, 29, 28], basé sur les intégrales de Sugeno (I.3) et (I.4). Rappelons que l'ensemble $\mathcal{S}$ (resp. $\mathcal{A}$) est comme dans l'introduction l'ensemble fini des états du système s (resp. des actions a). La variable représentant l'état du système est $S \in \mathcal{S}$. Soit $(\pi_a)_{a\in\mathcal{A}}$ une famille de distributions de possibilité sur $\mathcal{S}$, *i.e.* $\forall a \in \mathcal{A}$, $\pi_a(s) = \Pi_a(\{ S = s \})$ est le degré de possibilité de la situation $\{ S = s \} \subset \Omega$ lorsque l'action $a \in \mathcal{A}$ est sélectionnée. Soit $\rho : \mathcal{S} \to \mathcal{L}$ la fonction de préférence, définissant le degré de préférence de chaque état du système $s \in \mathcal{S}$.

Définition I.2.2 (*Critère de décision qualitatif*)

Soit π_a la distribution de possibilité décrivant l'incertitude à propos de l'état du système étant donné que l'action $a \in \mathcal{A}$ a été sélectionnée, et $\rho(s)$ la préférence de l'état du système $s \in \mathcal{S}$. En utilisant la formule (I.3) avec $f = \rho(S)$, l'intégrale de Sugeno de la préférence contre la mesure de possibilité Π_a mène à un critère optimiste pour $a \in \mathcal{A}$:

$$\mathbb{S}_{\Pi_a}[\rho(S)] = \max_{s\in\mathcal{S}} \min\{ \rho(s), \pi_a(s) \}. \tag{I.5}$$

De même, en utilisant la formule (I.4) avec $f = \rho(S)$, l'intégrale de Sugeno de la préférence contre la mesure de nécessité associée à Π_a, notée $\mathcal{N}_a$, mène au critère pessimiste pour $a \in \mathcal{A}$:

$$\mathbb{S}_{\mathcal{N}_a}[\rho(S)] = \min_{s\in\mathcal{S}} \max\{ \rho(s), 1 - \pi_a(s) \}. \tag{I.6}$$

Le critère pessimiste (I.6) cherche à éviter les états non désirés, tandis que le critère optimiste (I.5) souhaite rendre possible le fait d'atteindre les états préférés.

L'exemple suivant, illustré en figure I.3, montre bien la différence entre ces deux critères : soit $\mathcal{S} = \{ s_A, s_B, s_C \}$ l'ensemble des états et $\mathcal{A} = \{ a_1, a_2 \}$ l'ensemble des actions. Le modèle de préférence et le modèle d'incertitude sont décrits respectivement par ρ et $(\pi_a)_{a\in\mathcal{A}}$:

- $1 = \rho(s_A) > \rho(s_B) > \rho(s_C) = 0$;
- si l'action a_1 est sélectionnée, $\pi_{a_1}(s_A) = \pi_{a_1}(s_C) = 1$, et $\pi(s_B) = 0$;
- si l'action a_2 est sélectionnée, $\pi_{a_2}(s_A) = \pi(s_C) = 0$, et $\pi(s_B) = 1$, *i.e.* le système est dans l'état s_B de manière déterministe (avec nécessité 1).

Le critère optimiste est maximisé par l'action a_1, puisqu'avec cette action, le meilleur état du système, s_A, est entièrement possible. Cependant, cette action rend le pire état du système, s_C, entièrement possible, donc l'état s_A n'est pas du tout nécessaire : une action plus prudente est a_2, avec laquelle l'état devient s_B avec certitude, mais avec une plus petite préférence. On peut vérifier facilement que l'action a_2 maximise bien le critère pessimiste (I.6) :

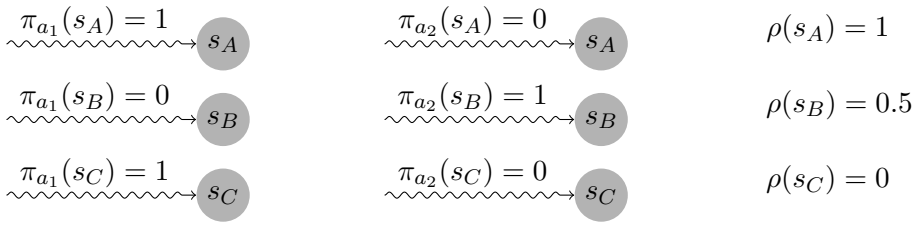


FIGURE I.3 – *Illustration de l'exemple de la section I.2 à propos des critères qualitatifs. L'action a_1 maximise le critère optimiste (I.5), qui peut mener au meilleur état (s_A), mais aussi au pire (s_C). Au contraire, l'action a_2 maximise le critère pessimiste (I.6) puisque le pire état n'est pas atteignable avec cette action.*

I.3 Processus Décisionnel Markovien Possibiliste Qualitatif (π-PDM)

Un *processus décisionnel markovien possibiliste qualitatif*, ou π-PDM, présenté dans [64, 63, 62], est la version possibiliste qualitative des PDM probabilistes décrits en introduction, basée sur les critères (optimistes et pessimistes) présentés en section I.2.

L'ensemble fini des états du système décrivant l'agent et son environnement, reste noté $\mathcal{S}$, comme vu en introduction avec les modèles probabilistes. L'ensemble fini des actions est toujours $\mathcal{A}$ et $\mathcal{L}$ est l'échelle possibiliste $\left\{0, \frac{1}{k}, \ldots, 1\right\}$, avec $k \geqslant 2$.

Comme dans le cas probabiliste, ce modèle considère que les états successifs du système, représentés par la séquence de variables $(S_t)_{t\in\mathbb{N}}$ avec $S_t \in \mathcal{S}\ \forall t \geqslant 0$, sont markovien. Dans ce cadre possibiliste qualitatif, cela signifie que la séquence $(S_t)_{t\in\mathbb{N}}$ est telle que $\forall t \geqslant 0, \forall (s_0, s_1, \ldots, s_{t+1}) \in \mathcal{S}^{t+2}$ et pour chaque séquence d'actions $(a_t)_{t\geqslant 0} \in \mathcal{A}^{\mathbb{N}}$, la variable S_{t+1} est indépendante (au sens causal) des variables $\{S_0, \ldots, S_{t-1}\}$, conditionnellement à $\{S_t = s\}$ et a_t :

$$\Pi\Big(S_{t+1} = s_{t+1} \;\Big|\; S_t = s_t, a_t \Big) = \Pi\Big(S_{t+1} = s_{t+1} \;\Big|\; S_t = s_t, S_{t-1} = s_{t-1}, \ldots, S_0 = s_0, (a_t)_{t\geqslant 0} \Big). \tag{I.7}$$

Cette indépendance possibiliste, ici causale, n'est pas la seule existante : une présentation générale des indépendances et des conditionnements, ainsi que de leurs conséquences dans les modèles graphiques, est disponible dans la thèse de N.Ben Amor [6].

En utilisant cette propriété markovienne, la dynamique du système est entièrement décrite avec les transitions possibilistes $\pi_t(s' \mid s, a) = \Pi\Big(S_{t+1} = s' \;\Big|\; S_t = s, a \Big) \in \mathcal{L} : \forall t \geqslant 0$, $(s, s') \in \mathcal{S}^2$ et $a \in \mathcal{A}$, $\pi_t(s' \mid s, a)$ est le degré de possibilité, qu'à l'étape de temps t, le système atteigne l'état s' lorsque l'agent choisit l'action a, conditionné au fait que l'état courant est s.

Enfin, un π-PDM est entièrement défini avec la séquence de fonctions de préférence $(\rho_t)_{t=0}^{H-1}$, où $\forall s \in \mathcal{S}$, $\forall a \in \mathcal{A}$, $\rho_t(s, a)$ est le degré de préférence lorsque l'état du système est s et l'agent sélectionne l'action a au temps t. La fonction de préférence terminale, Ψ, donne pour chaque état du système $s \in \mathcal{S}$, le degré de préférence de $S_H = s : \Psi(s)$.

Afin de définir les critères des π-PDM à partir des critères possibilistes (I.5) et (I.6), nous introduisons, pour un horizon $H \geqslant 0$, les *trajectoires de longueur H*, $\mathcal{T} = (s_1, \ldots, s_H)$, et $\mathcal{T}_H = \mathcal{S}^H$ l'ensemble de telles trajectoires. Une règle de décision est notée $\delta : \mathcal{S} \to \mathcal{L}$, et une stratégie de longueur H est une séquence de règles de décision $\delta_t : (\delta_t)_{t=0}^{H-1}$. L'ensemble de toutes les stratégies de taille H est noté Δ_H. Dans [61], pour une stratégie donnée $(\delta) \in \Delta_H$, une séquence d'états du système $\mathcal{T} = (s_1, \ldots, s_H) \in \mathcal{T}_H$, et un état initial donné $s_0 \in \mathcal{S}$, la *préference d'une stratégie de taille H commençant par s_0* est définie comme le degré de

possibilité le plus petit de la trajectoire et de s_0 :

$$\rho\Big(\mathcal{T},(\delta)\Big) = \min\left\{\min_{t=0}^{H-1}\rho\Big(s_t,\delta_t(s_t)\Big),\Psi(s_H)\right\}.$$

En utilisant la propriété de Markov de ce processus d'états du système, pour un état initial donné $s_0 \in \mathcal{S}$, un horizon $H \in \mathbb{N}$, et une stratégie $(\delta_t)_{t=0}^{H-1}$, le degré de possibilité de la trajectoire $\mathcal{T} = (s_1,\ldots,s_H)$ est

$$\Pi\left(S_H = s_h, S_{H-1} = s_{h-1},\ldots,S_1 = s_1 \middle| S_0 = s_0,(\delta_t)_{t=0}^{H-1}\right) = \min_{t=0}^{H-1}\pi_{t+1}\Big(s_{t+1}\Big|s_t,\delta_t(s_t)\Big) \quad \text{(I.8)}$$

noté $\pi\Big(\mathcal{T}\Big|s_0,(\delta)\Big)$.

L'intégrale de Sugeno de la préférence de la trajectoire contre cette distribution est notée

$$\mathbb{S}_\Pi\left[\rho\Big(\mathcal{T},(\delta)\Big)\;\middle|\; S_0 = s_0,(\delta)\right] = \mathbb{S}_\Pi\left[\min\left\{\min_{t=0}^{H-1}\rho\Big(S_t,\delta_t(S_t)\Big),\Psi(S_H)\right\}\;\middle|\; S_0 = s_0,(\delta)\right]$$

et correspond du critère optimiste definissant la stratégie optimale, *i.e.* une fonction valeur optimiste :

$$\overline{U_H}\Big(s_0,(\delta_t)_{t=0}^{H-1}\Big) = \max_{\mathcal{T}\in\mathcal{T}_H}\min\left\{\rho\Big(\mathcal{T},(\delta)\Big),\pi\Big(\mathcal{T}\Big|s_0,(\delta)\Big)\right\}. \quad \text{(I.9)}$$

C'est équivalent au critère optimiste (I.5), cependant, l'intégrale est sur les trajectoires $\mathcal{T}_H$, et la préférence dépend de la stratégie. La *stratégie optimale optimiste* $\overline{\delta^*}$ est la stratégie maximisant la fonction valeur optimiste (I.9), et la *fonction valeur optimiste optimale* est la fonction valeur optimiste la plus grande en faisant varier $(\delta) \in \Delta_H$:

Définition I.3.1 (*Fonction valeur et stratégie optimiste optimale*)

$\forall s \in \mathcal{S}$,

$$\overline{U_H^*}(s) = \max_{(\delta)\in\Delta_H}\left\{\overline{U_H}\Big(s,(\delta)\Big)\right\} \qquad \textit{(fonction valeur optimale optimiste)}, \quad \text{(I.10)}$$

$$\overline{\delta^*}(s) \in \operatorname*{argmax}_{(\delta)\in\Delta_H}\left\{\overline{U_H}\Big(s,(\delta)\Big)\right\} \qquad \textit{(stratégie optimale optimiste)}. \quad \text{(I.11)}$$

De même, le critère possibiliste qualitatif optimal (I.6) mène à un critère prudent pour les stratégies : la fonction valeur pessimiste est l'intégrale de Sugeno de la préférence de la trajectoire contre la mesure de nécessité qui vient de la distribution de possibilité sur les trajectoires $\mathcal{T}_H$ (I.8) avec la stratégie $(\delta) \in \Delta_H$:

$$\underline{U_H}\Big(s_0,(\delta_t)_{t=0}^{H-1}\Big) = \min_{\mathcal{T}\in\mathcal{T}_H}\max\left\{\rho\Big(\mathcal{T},(\delta)\Big),1-\pi\Big(\mathcal{T}\Big|s_0,(\delta)\Big)\right\}. \quad \text{(I.12)}$$

notée $\mathbb{S}_\mathcal{N}\left[\rho\Big(\mathcal{T},(\delta)\Big)\;\middle|\; S_0 = s,(\delta)\right]$. Comme précédemment pour le cas optimiste, la *stratégie optimale pessimiste* $\underline{\delta^*}$ est la stratégie maximisant la fonction valeur pessimiste (I.12), et la *fonction valeur pessimiste optimale* est la fonction valeur maximale en faisant varier la stratégie $(\delta) \in \Delta_H$:

Définition I.3.2 (*Fonction valeur et stratégie pessimiste optimale*)

$\forall s \in \mathcal{S}$,

$$\underline{U_H^*}(s) = \max_{(\delta)\in\Delta_H} \left\{ \underline{U_H}\Big(s,(\delta)\Big) \right\} \qquad \textit{(fonction valeur optimale pessimiste)}, \tag{I.13}$$

$$\underline{\delta^*}(s) \in \operatorname*{argmax}_{(\delta)\in\Delta_H} \left\{ \underline{U_H}\Big(s,(\delta)\Big) \right\} \qquad \textit{(stratégie optimale pessimiste)}. \tag{I.14}$$

Les fonctions valeur optimales et les stratégies peuvent être calculées par programmation dynamique :

Théoreme 2 (*Programmation Dynamique pour π-PDM*)

Le critère optimiste optimal et la stratégie optimale associée peuvent être calculés comme suit : $\forall s \in \mathcal{S}$,

$$\begin{aligned} &\overline{U_0^*}(s) = \Psi(s), \qquad \textit{and, } \forall 1 \leqslant i \leqslant H, \\ &\overline{U_i^*}(s) = \max_{a\in\mathcal{A}} \min \left\{ \rho_{H-i}(s,a), \max_{s'\in\mathcal{S}} \min \left\{ \pi_{H-i}\left(s' \mid s,a\right), \overline{U_{i-1}^*}(s') \right\} \right\}. \end{aligned} \tag{I.15}$$

$$\overline{\delta_{H-i}^*}(s) \in \operatorname*{argmax}_{a\in\mathcal{A}} \min \left\{ \rho_{H-i}(s,a), \max_{s'\in\mathcal{S}} \min \left\{ \pi_{H-i}\left(s' \mid s,a\right), \overline{U_{i-1}^*}(s') \right\} \right\}. \tag{I.16}$$

De même, le critère pessimiste optimal, et la stratégie associée peuvent être calculés comme suit : $\forall s \in \mathcal{S}$,

$$\begin{aligned} &\underline{U_0^*}(s) = \Psi(s), \qquad \textit{and, } \forall 1 \leqslant i \leqslant H, \\ &\underline{U_i^*}(s) = \max_{a\in\mathcal{A}} \min \left\{ \rho_{H-i}(s,a), \min_{s'\in\mathcal{S}} \max \left\{ 1-\pi_{H-i}\left(s' \mid s,a\right), \underline{U_{i-1}^*}(s') \right\} \right\}. \end{aligned} \tag{I.17}$$

$$\underline{\delta_{H-i}^*}(s) \in \operatorname*{argmax}_{a\in\mathcal{A}} \min \left\{ \rho_{H-i}(s,a), \min_{s'\in\mathcal{S}} \max \left\{ 1-\pi_{H-i}\left(s' \mid s,a\right), \underline{U_{i-1}^*}(s') \right\} \right\}. \tag{I.18}$$

Dans ce théorème, l'horizon i est l'opposé de l'étape du processus t modulo H : durant l'éxécution, $\delta_t = \delta_{H-i}$ est utilisée à l'étape de temps t, *i.e.* lorsque il reste i étapes.

Notons qu'une classe plus large de modèles PDM, incluant les PDM probabilistes et possibilistes, est appelé *PDM algébrique* [56].

La section suivante est dédiée à la présentation de l'homologue possibiliste qualitatif du PDMPO noté π-PDMPO : le modèle π-PDMPO est la version partiellement observable du modèle π-PDM. Ce modèle a été présenté dans [62] dans le cadre pessimiste. L'algorithme pour le résoudre a été présenté dans le cas où aucune préférence intermédiaire n'est impliquée *i.e.* dans le cas où les fonctions de préférence ρ_t ne sont pas prises en compte : dans ce cas, seule la fonction de préférence terminale Ψ modélise le but de la mission. Enfin, on remarque qu'il suffit de considérer un π-PDM classique avec $\rho_t(s,a) = 1$, $\forall s \in \mathcal{S}$, $\forall a \in \mathcal{A}$ et $\forall t \in \{0,\ldots,H-1\}$.

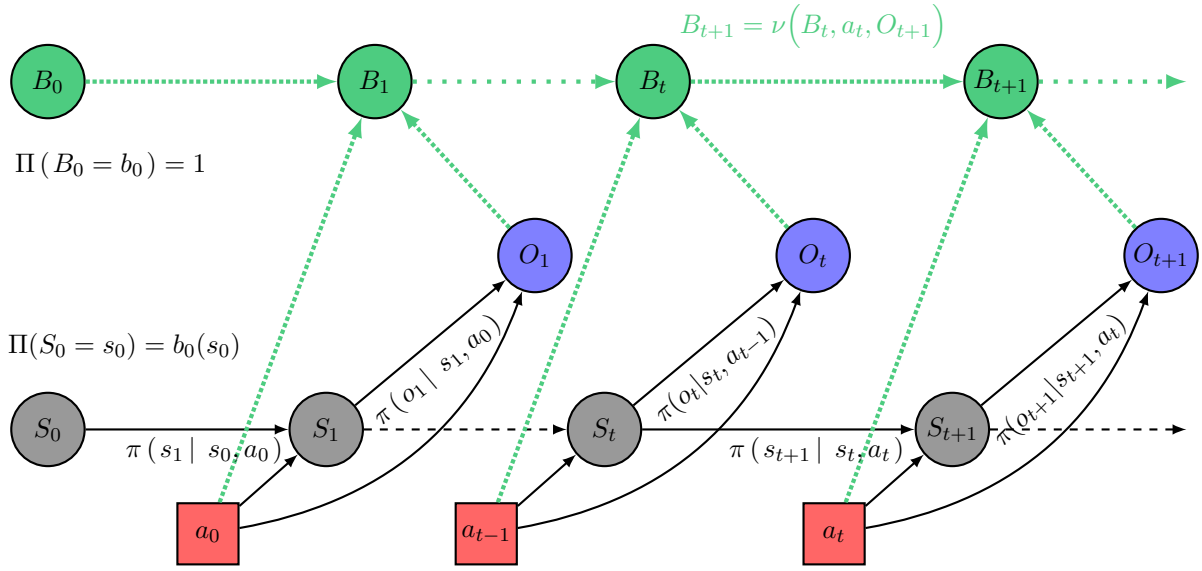


FIGURE I.4 – *Diagramme d'Influence d'un π-PDMPO et de son processus d'états de croyance : les ronds noirs représentent les états successifs du système S_t, les bleus représentent les observations successives O_t, et les carrés rouges sont les actions sélectionnées a_t. Les cercles verts en haut de la figure sont les états de croyance successifs B_t constituant le processus d'états de croyance, calculé avec la mise à jour $B_{t+1} = \nu(B_t, a_t, O_{t+1})$. Les lignes vertes en pointillé représentent des influences déterministes.*

I.4 PDM Partiellement Observable Possibiliste Qualitatif (π-PDMPO)

Le modèle PDMPO possibiliste qualitatif (π-PDMPO) a été présenté pour la première fois dans [62]. Comme pour le modèle probabiliste, dans le cadre partiellement observable, l'état du système n'est plus considéré comme une donnée d'entrée pour l'agent : l'agent doit l'estimer à partir des observations $o \in \mathcal{O}$ reçues à chaque étape de temps, représentées par le processus d'observation $(O_t)_{t\in\mathbb{N}}$. L'incertitude à propos des variables d'observation successives O_t ne dépend que de l'action et de l'état atteint : si l'agent choisit l'action $a \in \mathcal{A}$ au temps t, et le système a atteint l'état $s' \in \mathcal{S}$ à l'étape de temps $t+1$, l'observation $o' \in \mathcal{O}$ est reçue avec le degré de possibilité $\pi_t\,(o' \mid s', a\,) = \Pi\,(O_{t+1} = o' \mid S_{t+1} = s', a\,)$: conditionnellement à l'état suivant s' et à l'action courante a, la variable d'observation suivante est indépendante (au sens causal) de toutes les autres variables jusqu'à l'étape $t+1$. La figure I.4 illustre la dynamique et la structure d'un π-PDMPO. Un PDMPO probabiliste a la même structure, sauf que les distributions de possibilité de transition (resp. d'observation) $\pi\,(s' \mid s, a\,)$ (resp. $\pi\,(o' \mid s', a\,)$). doivent être remplacée par la distribution de probabilité $\mathbf{p}\,(s' \mid s, a\,)$ (resp. $\mathbf{p}\,(o' \mid s', a\,)$).

Comme avec le modèle probabiliste, le calcul des stratégies est effectué en traduisant le π-PDMPO en un π-PDM entièrement observable. L'espace d'état de ce dernier est l'ensemble des *états de croyance possibilistes qualitatifs* $\beta : \mathcal{S} \to \mathcal{L}$ décrivant la connaissance à propos de l'état du système, *i.e.* l'ensemble de distributions de possibilité sur $\mathcal{S}$. Cet ensemble est noté $\Pi_{\mathcal{L}}^{\mathcal{S}} = \{\,\pi : \mathcal{S} \to \mathcal{L} \mid \max_{s\in\mathcal{S}} \pi(s) = 1\,\}$. Notons tout d'abord que le nombre d'états de croyance à propos de l'état du système est

$$(\#\mathcal{L})^{\#\mathcal{S}} - (\#\mathcal{L} - 1)^{\#\mathcal{S}}\,. \tag{I.19}$$

En effet, il y a $\#\mathcal{L}^{\#\mathcal{S}}$ fonctions différentes de $\mathcal{S}$ vers $\mathcal{L}$, et $(\#\mathcal{L} - 1)^{\#\mathcal{S}}$ fonctions non normalisées *i.e.* fonctions $f : \mathcal{S} \to \mathcal{L}$ telles que $\max_{s\in\mathcal{S}} f(s) < 1$. Le nombre de distributions de

possibilité sur $\mathcal{S}$ est le nombre de fonctions normalisées de $\mathcal{S}$ vers $\mathcal{L}$, *i.e.* le nombre total de fonctions, moins le nombre de fonctions non normalisées.

Tout d'abord, définissons formellement un π-PDMPO comme le 7-uplet $< \mathcal{S}, \mathcal{A}, \mathcal{O}, T^\pi, O^\pi, \Psi, \beta_0 >$:

- $\mathcal{S}$, un ensemble fini d'états (cachés) du système ;
- $\mathcal{A}$ un ensemble fini d'actions ;
- $\mathcal{O}$ un ensemble fini d'observations ;
- T^π l'ensemble des distributions de possibilité de transition $\pi(s' \mid s, a)$;
- O^π, l'ensemble des distributions de possibilité d'observation $\pi(o' \mid s', a)$;
- Ψ la fonction de préférence, définissant pour chaque état $s \in \mathcal{S}$, la préférence associée à la situation où l'état du système est $s \in \mathcal{S}$;
- β_0, *l'état de croyance possibiliste initial*, est la distribution de possibilité définissant l'incertitude à propos de l'état initial : $\forall s \in \mathcal{S}$, $\beta_0(s) = \Pi(S_0 = s)$.

À chaque étape de temps, l'état de croyance possibiliste est calculé à partir de ces objets : 'état de croyance initial $\beta_0 \in \Pi_{\mathcal{L}}^{\mathcal{S}}$ fait partie de la définition du π-PDMPO.

À l'étape de temps $t \geqslant 1$, l'état de croyance est la distribution de possibilité sur l'état courant du système, conditionnellement à toutes les données visibles par l'agent.

Définition I.4.1 (*État de croyance possibiliste qualitatif*)

$$\beta_t(s) = \Pi(S_t = s \mid O_1 = o_1, \ldots, O_t = o_t, a_0, \ldots, a_{t-1}) = \Pi(S_t = s \mid I_t = i_t) \tag{I.20}$$

où $i_t = \{o_1, \ldots, o_t, a_0, \ldots, a_{t-1}\}$ est l'information visible par l'agent à l'étape de temps t ($i_0 = \{\} = \emptyset$), et I_t la variable correspondante.

La mise à jour possibiliste de l'état de croyance est basée sur l'homologue possibiliste de la règle de Bayes :

Théoreme 3 (*Mise à jour possibiliste qualitative de l'état de croyance*)

Si l'état de croyance à l'étape de temps t est β_t, l'action choisie est $a_t \in \mathcal{A}$, et l'état suivant est o_{t+1}, l'état suivant β_{t+1} est calculé comme suit :

$$\beta_{t+1}(s') = \begin{cases} 1 & \text{si } \pi_t(s', o_{t+1} \mid \beta_t, a_t) = \pi_t(o_{t+1} \mid \beta_t, a_t), \\ \pi_t(s', o_{t+1} \mid \beta_t, a_t) & \text{sinon.} \end{cases} \tag{I.21}$$

où la distribution de possibilité jointe sur la variable d'état S_{t+1} et la variable d'observation O_{t+1} conditionnellement à l'information courante, est notée $\pi_t(s', o' \mid \beta_t, a_t) = \min\left\{\pi_t(o' \mid s', a_t), \max_{s \in \mathcal{S}} \min\left\{\pi_t(s' \mid s, a_t), \beta_t(s)\right\}\right\}$. La notation $\pi(o' \mid \beta_t, a_t)$ est aussi utilisée pour $\max_{s' \in \mathcal{S}} \pi_t(s', o' \mid \beta_t, a_t)$.

Cette formule est appelée la ***mise à jour de la croyance possibiliste****, et puisque l'état de croyance β_{t+1} est une fonction de β_t, a_t and o_{t+1}, nous notons cette mise à jour*

$$\beta_{t+1} = \nu(\beta_t, a_t, o_{t+1}),$$

avec ν appelée fonction de mise à jour.

La mise à jour possibiliste de l'état de croyance (I.21) est notée

$$\beta_{t+1}(s') \propto^\pi \pi_t(s', o_{t+1} \mid \beta_t, a_t)$$

puisqu'elle consiste seulement à normaliser la fonction $s' \mapsto \pi(s', o_{t+1} \mid \beta_t, a_t)$ au sens possibiliste ($\max_s \pi(s) = 1$).

Nous notons B_t^π l'état de croyance lorsque considéré comme une variable : B_0^π est déterministe égal à β_0 (mais S_0 est incertain, ce qui est décrit par la distribution de possibilité β_0) et $B_{t+1}^\pi = \nu(B_t^\pi, a_t, O_{t+1})$ où O_{t+1} est la variable d'observation à l'étape de temps $t+1$.

Afin de rendre les choses plus claires, le modèle π-PDMPO est ici défini sans préférences intermédiaires ρ_t, mais avec une préférence terminale Ψ uniquement.

Nous pouvons maintenant exprimer la distribution de possibilité de transition du processus de croyance *i.e.* les éléments de $\tilde{T}$, comme suit : $\forall t \geqslant 0$,

$$\pi_t\left(\beta' \mid \beta, a\right) = \max_{\substack{o' \in \mathcal{O} \text{ s.t.} \\ \nu(\beta,a,o')=\beta'}} \pi_t\left(o' \mid \beta, a\right), \tag{I.22}$$

où $\pi_t\left(o' \mid \beta, a\right) = \max_{(s,s') \in \mathcal{S}^2} \min\Big\{\pi_t\left(o' \mid s', a_t\right), \pi_t\left(s' \mid s, a_t\right), \beta(s)\Big\}$, est le degré de possibilité d'observer o' conditionnellement à toutes les informations précédentes.

Enfin, la fonction de préférence associée à l'état de croyance possibiliste β_H est défini de manière pessimiste : $\forall \beta \in \Pi_{\mathcal{L}}^{\mathcal{S}}$,

$$\underline{\Psi}(\beta) = \min_{s \in \mathcal{S}} \max\left\{\Psi(s), 1 - \beta(s)\right\}. \tag{I.23}$$

Le cas optimiste donne une préférence maximale à l'ignorance totale, ce qui ne semble pas pouvoir mener l'agent à estimer l'état du système et à atteindre un état satisfaisant.

Il est possible de montrer qu'il est suffisant de chercher une stratégie basée sur les états de croyance *i.e.* parmi les suites de règles de décision $(\delta_t)_{t=0}^{H-1}$, telles que $\forall t \geqslant 0$, $\delta_t : \beta_t \mapsto \delta_t(\beta_t) \in \mathcal{A}$.

Le π-PDM $\langle \tilde{\mathcal{S}}^\pi, \mathcal{A}, \tilde{T^\pi}, \underline{\Psi} \rangle$ construit à partir d'un π-PDMPO $\langle \mathcal{S}, \mathcal{A}, \mathcal{O}, T^\pi, O^\pi, \beta_0 \rangle$ est finalement :

- $\tilde{\mathcal{S}}^\pi = \Pi_{\mathcal{L}}^{\mathcal{S}}$, l'ensemble de tous les états de croyance (possibilistes qualitatifs) possibles ;
- $\tilde{T^\pi}$ contient toutes les distributions de possibilité de transition sur les états de croyance : $\forall a \in \mathcal{A}$, $\forall \beta \in \Pi_{\mathcal{L}}^{\mathcal{S}}$, ces distributions de possibilité sont définies par l'équation (I.22), $\pi_t\left(. \mid \beta, a\right)$ is in $\tilde{T^\pi}$;
- la fonction de préférence est l'estimation pessimiste de la préférence : $\underline{\Psi}$, *cf.* équation (I.23).

Notons maintenant que, en utilisant la définition de la fonction de transition des états de croyance (I.22), pour chaque fonction de l'espace des états de croyance vers $\mathcal{L}$, $U : \Pi_{\mathcal{L}}^{\mathcal{S}} \to \mathcal{L}$,

$$\begin{aligned}
\max_{\beta' \in \Pi_{\mathcal{L}}^{\mathcal{S}}} \min\left\{\pi_t\left(\beta' \mid \beta, a\right), U(\beta')\right\} &= \max_{\beta' \in \Pi_{\mathcal{L}}^{\mathcal{S}}} \min\left\{\max_{\substack{o' \in \mathcal{O} \text{ s.t.} \\ \nu(\beta,a,o')=\beta'}} \pi_t\left(o' \mid \beta, a\right), U(\beta')\right\} \\
&= \max_{\beta' \in \Pi_{\mathcal{L}}^{\mathcal{S}}} \max_{\substack{o' \in \mathcal{O} \text{ s.t.} \\ \nu(\beta,a,o')=\beta'}} \min\left\{\pi_t\left(o' \mid \beta, a\right), U(\beta')\right\} \\
&= \max_{o' \in \mathcal{O}} \min\Big\{\pi_t\left(o' \mid \beta, a\right), U\Big(\nu(\beta, a, o')\Big)\Big\},
\end{aligned}$$

Cette observation mène à l'algorithme 1 qui est l'application directe du schéma de programmation dynamique optimiste pour π-PDM (théorème 2) avec seulement une préférence terminale (qui est pessimiste) : ce schéma est utilisé sur le π-PDM $\langle \tilde{\mathcal{S}}^\pi, \mathcal{A}, \tilde{T^\pi}, \underline{\Psi} \rangle$.

L'algorithme 2, qui est la programmation dynamique pessimiste pour π-PDM (théorème 2), avec une préférence terminale seulement : elle est appliquée au π-MDP $\langle \tilde{\mathcal{S}}^\pi, \mathcal{A}, \tilde{T^\pi}, \underline{\Psi} \rangle$.

Dans ce premier chapitre la théorie des possibilités qualitatives ainsi que les modèle π-PDM et π-PDMPO ont été présentés.

Dans le cadre probabiliste des PDMPO, l'incertitude est décrite avec des probabilités $\mathbf{p}\left(s' \mid s, a\right) \in \mathbb{R}$ et $\mathbf{p}\left(o' \mid s', a\right) \in \mathbb{R}$ tandis qu'elle est définie par des distributions de possibilité

Algorithm 1: Algorithme de programmation dynamique pour π-PDMPO optimiste avec une préférence sur l'état final seulement

$\overline{U_0^*} \leftarrow \underline{\Psi}$;
for $i \in \{1, \ldots, H\}$ **do**
  for $\beta \in \Pi_{\mathcal{L}}^{\mathcal{S}}$ **do**
    $\overline{U_i^*}(\beta) \leftarrow \max_{a \in \mathcal{A}} \max_{o' \in \mathcal{O}} \min \left\{ \pi_t\left(o' \mid \beta, a\right), \overline{U_{i-1}^*}\Big(\nu(\beta, a, o')\Big) \right\}$;
    $\overline{\delta_{H-i}}(\beta) \in \operatorname*{argmax}_{a \in \mathcal{A}} \max_{o' \in \mathcal{O}} \min \left\{ \pi_t\left(o' \mid \beta, a\right), \overline{U_{i-1}^*}\left(\nu(\beta, a, o')\right) \right\}$;
return $\overline{U_H^*}$, $(\overline{\delta^*})$;

Algorithm 2: Algorithme de programmation dynamique pour π-PDMPO pessimiste avec une préférence terminale seulement

$\underline{U_0^*} \leftarrow \underline{\Psi}$;
for $i \in \{1, \ldots, H\}$ **do**
  for $\beta \in \Pi_{\mathcal{L}}^{\mathcal{S}}$ **do**
    $\underline{U_i^*}(\beta) \leftarrow \max_{a \in \mathcal{A}} \min_{o' \in \mathcal{O}} \max \left\{ 1 - \pi_t\left(o' \mid \beta, a\right), \underline{U_{i-1}^*}\Big(\nu(\beta, a, o')\Big) \right\}$;
    $\underline{\delta_{H-i}}(\beta) \in \operatorname*{argmax}_{a \in \mathcal{A}} \min_{o' \in \mathcal{O}} \max \left\{ 1 - \pi_t\left(o' \mid \beta, a\right), \underline{U_{i-1}^*}\left(\nu(\beta, a, o')\right) \right\}$;
return $\underline{U_H^*}$, $(\underline{\delta^*})$;

$\pi\left(s' \mid s, a\right) \in \mathcal{L} = \left\{0, \frac{1}{k}, \ldots, 1\right\}$ (avec $k \geqslant 1$) et $\pi\left(o' \mid s', a\right) \in \mathcal{L}$ dans le cadre π-PDMPO. De plus, le cadre probabiliste mesure l'intérêt de passer par un état $s \in \mathcal{S}$ et d'utiliser l'action $a \in \mathcal{A}$ avec la fonction de récompense $r(s, a) \in \mathbb{R}$ tandis que le cadre possibiliste qualitatif utilise des préférences $\rho(s, a) \in \mathcal{L}$ et $\Psi(s) \in \mathcal{L}$. Ainsi, le critère probabiliste pour une stratégie donnée, est l'espérance des récompenses, ce qui s'écrit $\mathbb{E}\left[rewards\Big((S_t)_{t \geqslant 0}\Big)\right] \in \mathbb{R}$: dans le cadre possibiliste, deux critères sont possibles, qui sont les intégrales de Sugeno de la préférence, $\mathbb{S}_{\Pi}\left[preferences\Big((S_t)_{t \geqslant 0}\Big)\right] \in \mathcal{L}$ pour l'optimiste, et $\mathbb{S}_{\mathcal{N}}\left[preferences\Big((S_t)_{t \geqslant 0}\Big)\right] \in \mathcal{L}$ pour la pessimiste.

Un PDMPO (resp. π-PDMPO), est redéfini en termes de PDM (resp. π-PDM) entièrement observable où l'état du système est l'état de croyance $b_t \in \mathbb{P}_{b_0}^{\mathcal{S}}$ (resp. $\beta_t \in \Pi_{\mathcal{L}}^{\mathcal{S}}$), *i.e.* les distribution de probabilité (resp. possibilité) sur les états du système du PDMPO : la récompense basée sur la croyance est définie comme étant $r(b, a) = \mathbb{E}_{S \sim b}\left[r(S, a)\right] = \sum_{s \in \mathcal{S}} r(s, a) \cdot b(s)$ dans le cas probabiliste. Dans le cas possibiliste qualitatif, la préférence basée sur l'état de croyance est $\underline{\rho}(b, a) = \mathbb{S}_{\mathcal{N}, S \sim \beta}\left[\rho(S, a)\right] = \min_{s \in \mathcal{S}} \max\left\{\rho(s, a), 1 - \beta(s)\right\}$ pour les pessimistes.

Le chapitre suivant propose certaines amélioration au modèle possibiliste qualitatif : La propriété d'observabilité mixte est définie : comme pour le modèle probabiliste, la complexité de résolution des problèmes ayant cette propriété est réduite. Enfin, l'homologue du problème à horizon infini est formellement défini, et il est prouvé que l'algorithme de résolution proposé pour le résoudre renvoie une stratégie optimale.

Mises à Jour et Étude Pratique des π-PDMPO

II

La fin du chapitre précédent a présenté le modèle π-PDMPO, l'homologue possibiliste qualitatif des PDMPO probabilistes. Rappelons que, dans le cadre possibiliste qualitatif, l'ensemble des états de croyance est fini, $\#\Pi_{\mathcal{L}}^{\mathcal{S}} < +\infty$ (*cf.* équation I.19) tandis que l'ensemble des états de croyance est infini dans le cadre probabiliste $\mathbb{P}_{b_0}^{\mathcal{S}}$: pour cette raison, les π-PDMPO peuvent être vus comme un modèle plus simple que les PDMPO probabilistes pour la décision séquentielle dans l'incertain avec observabilité partielle.

Ce modèle peut simplifié de plus belle, lorsque le problème satisfait la propriété d'*observabilité mixte*, comme montré dans la section qui suit.

II.1 Observabilité Mixte et π-PDM à Observabilité Mixte (π-PDMOM)

La résolution d'un π-PDMPO se confronte au fait que la taille de l'espace des états de croyance $\Pi_{\mathcal{L}}^{\mathcal{S}}$ est exponentielle en fonction de la taille de l'espace des états du système $\mathcal{S}$, *cf.* équation (I.19) de la section I.4. Cependant, en pratique, les états du système sont rarement totalement cachés. Utiliser la propriété d'observabilité mixte peut être une solution : inspiré par

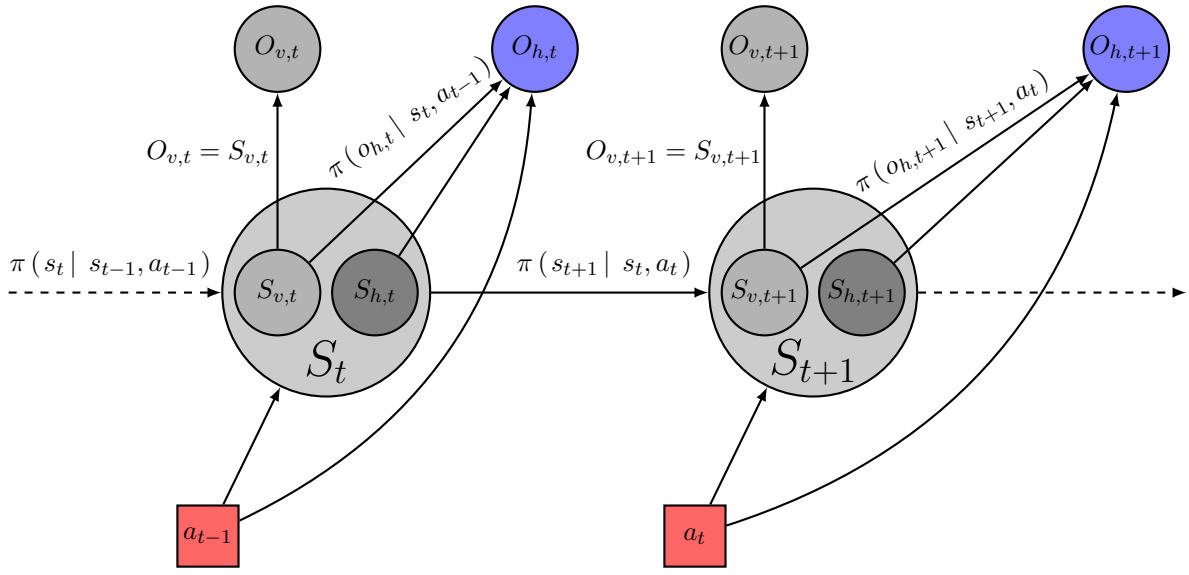


Figure II.1 – *Réseau bayésien dynamique d'un π-PDMOM : à l'étape de temps t, l'état du système est décrit par la variable $S_t = (S_{v,t}, S_{h,t})$. L'observation reçue est $O_t = (O_{v,t}, O_{h,t})$ avec $O_{v,t} = S_{v,t}$, et $O_{h,t}$ dépend de S_t et de l'action a_t.*

un travail récent dans le cadre des PDMPO probabilistes, [52, 2], nous présentons dans cette section un modèle structuré qui prend en compte les situations dans lesquelles l'agent observe directement une partie de l'état du système. Un π-PDMPO qui modélise une telle situation respecte la *propriété d'observabilité mixte*. Les états de croyance ne sont alors utilisés que pour les composantes partiellement observables et la taille de l'espace d'état est significativement résuite. Ainsi, ce modèle généralise les π-PDM et les π-PDMPO.

Comme dans [2], nous faisons l'hypothèse que l'espace d'état $\mathcal{S}$ d'un PDMOM possibiliste qualitatif (π-PDMOM) peut être écrit comme le produit cartésien d'un espace d'états visibles $\mathcal{S}_v$ et d'un espace d'états cachés $\mathcal{S}_h$: $\mathcal{S} = \mathcal{S}_v \times \mathcal{S}_h$. Soit $s = (s_v, s_h)$ un état du système. La composante s_v est directement observée par l'agent et s_h est seulement observé partiellement à travers les observations de l'ensemble $\mathcal{O}_h$: nous notons $\pi_t(o'_h \mid s', a)$, la distribution de possibilité sur l'observation suivante $o'_h \in \mathcal{O}_h$ à l'étape de temps t, connaissant l'état suivant $s' \in \mathcal{S}$ et l'action courante $a \in \mathcal{A}$. La figure II.1 illustre la structure de ce modèle à observabilité mixte.

L'espace d'état visible est identifié à l'espace des observations : $\mathcal{O}_v = \mathcal{S}_v$ et $\mathcal{O} = \mathcal{O}_v \times \mathcal{O}_h$. Ainsi, sachant que la composante visible de l'état est s_v, l'agent observe *nécessairement* $o_v = s_v$ ($\forall a \in \mathcal{A}$, si $o'_v \neq s_v$, $\pi_t(o'_v \mid s'_v, a) = 0$). Formellement, vu comme un π-PDMPO, sa distribution de possibilité sur les observations peut s'écrire :

$$\begin{aligned}
\pi_t(o' \mid s', a) &= \pi_t(o'_v, o'_h \mid s'_v, s'_h, a) \\
&= \min\{\pi_t(o'_h \mid s'_v, s'_h, a), \pi_t(o'_v \mid s'_v)\} \\
&= \begin{cases} \pi_t(o'_h \mid s', a) & \text{if } o'_v = s'_v \\ 0 & \text{sinon} \end{cases}, \qquad \text{(II.1)}
\end{aligned}$$

puisque $\forall a \in \mathcal{A}$, $\pi_t(o'_v \mid s'_v) = 1$ si $s'_v = o'_v$ et 0 sinon. Le théoreme suivant, basé sur cette égalité, permet de définir les états de croyance sur les états cachés du système.

Théoreme 4 (*Nature des états de croyance atteignables*)

Chaque état de croyance atteignable d'un π-PDMOM peut être écrit comme un élément de $\mathcal{S}_v \times \Pi_{\mathcal{L}}^{\mathcal{S}_h}$ où $\Pi_{\mathcal{L}}^{\mathcal{S}_h}$ est l'espace des distributions de possibilité sur $\mathcal{S}_h$: tout $\beta \in \Pi_{\mathcal{L}}^{\mathcal{S}}$ atteignable peut s'écrire (s_v, β_h) avec $\beta_h(s_h) = \max_{\overline{s}_v \in \mathcal{S}_v} \beta(\overline{s}_v, s_h)$ et $s_v = \operatorname{argmax}_{\overline{s}_v \in \mathcal{S}_v} \beta(\overline{s}_v, s_h)$.

Comme tous les états de croyance sont dans $\mathcal{S}_v \times \Pi_{\mathcal{L}}^{\mathcal{S}_h}$ lorsque le π-PDMPO satisfait la propriété d'observabilité mixte, le théorème suivant réécrit la mise à jour du nouvel état de croyance $\beta_h \in \Pi_{\mathcal{L}}^{\mathcal{S}}$ *i.e.* de l'état de croyance sur les états cachés du système $s_h \in \mathcal{S}_h$.

Théoreme 5 (*Mise à jour de la croyance pour un π-PDMOM*)

Si un problème peut se modéliser par un π-PDMOM

$$\Big\langle S_v \times S_h, \mathcal{A}, \mathcal{O}_h, T^\pi, O^\pi, \Psi, \beta_0 = (s_{v,0}, \beta_{h,0}) \Big\rangle,$$

une nouvelle fonction de mise à jour de l'état de croyance ν_h peut être définie : si, à l'étape de temps t, l'état visible courant est $s_{v,t} \in \mathcal{S}_v$, l'état de croyance courant sur l'état caché est $\beta_{h,t} \in \Pi_{\mathcal{L}}^{\mathcal{S}_h}$, l'action choisie courante est $a_t \in \mathcal{A}$, l'état visible suivant est $s_{v,t+1} \in \mathcal{S}_v$ et l'observation suivante est $o_{h,t+1} \in \mathcal{O}_h$, alors l'état de croyance suivant à propos de l'état caché du système est

$$\beta_{h,t+1}(s'_h) = \begin{cases} 1 & \text{si } \begin{array}{l} \pi_t(s'_h, s_{v,t+1}, o_{h,t+1} \mid s_{v,t}, \beta_{h,t}, a_t) \\ = \pi_t(s_{v,t+1}, o_{h,t+1} \mid s_{v,t}, \beta_{h,t}, a_t) \end{array} \\ \pi_t(s'_h, s_{v,t+1}, o_{h,t+1} \mid s_{v,t}, \beta_{h,t}, a_t) & \text{sinon} \end{cases}, \qquad \text{(II.2)}$$

où

$$\pi_t\left(s'_v, s'_h, o'_h \mid\ s_v, \beta_h, a\right) = \min\left\{\pi_t\left(o'_h \mid\ s', a\right), \max_{s_h \in \mathcal{S}_h} \min\left\{\pi_t\left(s' \mid\ s_v, s_h, a\right), \beta_h(s_h)\right\}\right\}$$

est la distribution de possibilité jointe sur les états du système $s'_h \in \mathcal{S}_h$ et les objets visibles (état visible du système et observation) $s'_v \in \mathcal{S}_v$ et $o'_h \in \mathcal{O}_h$. La mise à jour de l'état de croyance est notée

$$\beta'_h = \nu_h\left(s_v, \beta_h, a, s'_v, o'_h\right).$$

L'espace d'état du π-PDM résultant d'un π-PDMOM peut alors être restreint à l'espace produit $\mathcal{S}_v \times \Pi_{\mathcal{L}}^{\mathcal{S}_h}$, *i.e.* un espace d'état plus petit grâce à l'observabilité mixte : le π-PDM résultant est $\langle \tilde{S^\pi}, \tilde{T^\pi}, \mathcal{A}, \tilde{\Psi}\rangle$, où

— l'espace des états du système est $\tilde{S^\pi} = \mathcal{S}_v \times \Pi_{\mathcal{L}}^{\mathcal{S}_h}$,

— la distribution de probabilité de transition dans $\tilde{T^\pi}$ est telle que $\forall\{0, \ldots, H-1\}$, $\forall a \in \mathcal{A}$, $\forall\Big[(s_v, \beta_h), (s'_v, \beta'_h)\Big] \in \left(\tilde{S^\pi}\right)^2$,

$$\pi_t\Big((s'_v, \beta'_h)\Big|(s_v, \beta_h), a\Big) = \max_{\substack{o'_h \in \mathcal{O}_h \text{ s.t.} \\ \nu_h(s_v, \beta_h, a, s'_v, o'_h) = \beta'_h}} \pi_t\left(s'_v, o'_h \mid\ s_v, \beta_h, a\right),$$

où $\pi_t\left(s'_v, o'_h \mid\ s_v, \beta_h, a\right)$ est défini dans le théorème au-dessus,

— la préférence pessimiste *i.e.* $\tilde{\Psi} = \underline{\Psi}$: elle peut être réécrite $\forall s_v \in \mathcal{S}_v, \forall \beta_h \in \Pi_{\mathcal{L}}^{\mathcal{S}_h}, \forall a \in \mathcal{A}$,

$$\underline{\Psi}(s_v, \beta_h) = \min_{s_h \in \mathcal{S}_h} \max\left\{\Psi(s_v, s_h), 1 - \beta_h(s_h)\right\}.$$

Un algorithme standard aurait calculé la fonction valeur pour chaque $\beta \in \Pi_{\mathcal{L}}^{\mathcal{S}}$ tandis que l'équation de programmation dynamique du π-PDM résultant mène à un algorithme qui la calcule seulement pour chaque $(s_v, \beta_h) \in \mathcal{S}_v \times \Pi_{\mathcal{L}}^{\mathcal{S}_h}$, puisque seuls ces états de croyance sont atteignables. La taille du nouvel espace des états de croyance est

$$\#(\mathcal{S}_v \times \Pi_{\mathcal{L}}^{\mathcal{S}_h}) = \#\mathcal{S}_v \times \left(\#\mathcal{L}^{\#\mathcal{S}_h} - (\#\mathcal{L} - 1)^{\#\mathcal{S}_h}\right),$$

ce qui est exponentiellement plus petit que la taille de l'espace des états de croyance du π-PDMPO équivalent :

$$\#\Pi_{\mathcal{L}}^{\mathcal{S}} = \#\mathcal{L}^{\#\mathcal{S}_v \times \#\mathcal{S}_h} - (\#\mathcal{L} - 1)^{\#\mathcal{S}_v \times \#\mathcal{S}_h}.$$

II.2 Horizon Indéterminé

Pour de nombreux problèmes en pratique, il est très difficile de déterminer un horizon H. Le but de cette section est de présenter un algorithme pour résoudre les π-PDMOM avec préférence terminale, et horizon indéterminé.

À notre connaissance, nous proposons ici le premier algorithme d'itération sur les valeurs (IV) pour π-PDM qui renvoie une stratégie optimale pour un critère spécifié. Comme mentionné dans [62], nous faisons l'hypothèse de l'existence d'une action "rester", notée $\widehat{a}$, qui retient le système dans son état courant avec nécessité 1. Cette action est l'homologue possibiliste du facteur d'actualisation γ dans le modèle probabiliste, puisqu'il garantit la convergence de l'algorithme d'itération sur les valeurs. Nous verrons cependant que l'action $\widehat{a}$ n'est finalement utilisée que sur certains états buts. Notons qu'une hypothèse similaire est utilisée pour calculer des stratégies optimales dans le cadre des processus déterministes (planification classique) dont l'horizon n'est pas spécifié [44].

Algorithm 3: Algorithme IV pour π-PDM – Préférence Terminale

for $s \in \mathcal{S}$ **do**
  $\overline{U^*}(s) \leftarrow 0$;
  $\overline{U^c}(s) \leftarrow \Psi(s)$;
  $\overline{\delta^*}(s) \leftarrow \widehat{a}$;
while $\overline{U^*} \neq \overline{U^c}$ **do**
  $\overline{U^*} = \overline{U^c}$;
  for $s \in \mathcal{S}$ **do**
    $\overline{U^c}(s) \leftarrow \max_{a \in \mathcal{A}} \max_{s' \in \mathcal{S}} \min \left\{ \pi\left(s' \mid s, a\right), \overline{U^*}(s') \right\}$;
    if $\overline{U^c}(s) > \overline{U^*}(s)$ **then**
      $\overline{\delta^*}(s) \in \operatorname{argmax}_{a \in \mathcal{A}} \max_{s' \in \mathcal{S}} \min \left\{ \pi\left(s' \mid s, a\right), \overline{U^*}(s') \right\}$;
return $\overline{U^*}$, $\overline{\delta^*}$;

Nous notons $\widehat{\delta}$ la règle de décision telle que $\forall s \in \mathcal{S}$, $\widehat{\delta}(s) = \widehat{a}$. L'ensemble fini de toutes les statégies est $\Delta = \cup_{i \geqslant 1} \Delta_i$, et $\#\delta$ est la taille de la stratégie (δ) en termes d'étapes de décision. Nous pouvons maintenant définir le critère optimiste pour un horizon indéterminé : si $(\delta) \in \Delta$,

$$\overline{U}\Big(s_0, (\delta)\Big) = \max_{\mathcal{T} \in \mathcal{T}_{\#\delta}} \min \left\{ \pi\Big(\mathcal{T}\Big| s_0, (\delta)\Big), \Psi(s_{\#\delta}) \right\}, \tag{II.3}$$

où $\mathcal{T} = (s_1, \ldots, s_{\#\delta})$ est une trajectoire d'états du système, $\mathcal{T}_{\#\delta}$ l'ensemble de telles trajectoires, et

$$\pi\Big(\mathcal{T}\Big| s_0, (\delta)\Big) = \min_{i=0}^{\#\delta - 1} \pi\Big(s_{i+1}\Big| s_i, \delta_i(s_i)\Big).$$

Théoreme 6 (*Optimalité de l'algorithme d'IV pour les π-PDM optimistes*)

Si il existe une action $\widehat{a}$ telle que, pour chaque $s \in \mathcal{S}$, $\pi\left(s' \mid s, \widehat{a}\right) = 1$ si $s' = s$ et 0 sinon, alors l'algorithme 3 calcule le critère maximal et une stratégie optimale, i.e. qui maximise le critère (II.3), et qui est stationnaire (i.e. qui ne dépend pas de l'étape du processus t).

Soit s un état tel que $\overline{\delta^*}(s) = \widehat{a}$, où $\overline{\delta^*}$ est la stratégie renvoyée par l'algorithme. En regardant l'algorithme 3, nous pouvons remarquer que $\overline{U^*}(s)$ reste égal à $\Psi(s)$ durant les itérations de l'algorithme après le premier passage dans la boucle while. Donc, $\forall s' \in \mathcal{S}$, soit $\forall a \in \mathcal{A}$, $\Psi(s) \geqslant \pi\left(s' \mid s, a\right)$, soit $\Psi(s) \geqslant \overline{U^*}(s')$. Si le problème n'est pas trivial, cela signifie que s est un but $(\Psi(s) > 0)$ et que les degrés de possibilité de transition vers de meilleurs buts sont plus petit que le degré de préférence de s.

II.3 Résultats Expérimentaux

Considérons un robot sur une grille de taille $g \times g$, avec $g > 1$. Il connait parfaitement sa position sur la grille $(x, y) \in \{1, \ldots, g\}^2$ à chaque étape du processus, ce qui constitue l'espace des états visibles $\mathcal{S}_v$. Il se trouve initialement à la position $s_{v,0} = (1, 1)$. Deux cibles immobiles sont présentes sur la grille : la "cible 1" est en $(x_1, y_1) = (1, g)$, la "cible 2" se trouve en $(x_2, y_2) = (g, 1)$ sur la grille, et le robot connait parfaitement leurs positions. Une des deux cibles est A, l'autre est B, et la mission du robot est d'identifier et d'atteindre A aussi tôt que possible. Le robot ne sait pas quelle cible est A : les deux situations A_1 et A_2 correspondent respectivement à "la cible 1 est A" et "la cible 2 est A" et constituent l'espace des états cachés

$\mathcal{S}_h$. Les actions $\mathcal{A}$ sont les déplacements dans les quatre directions ainsi que l'action "rester" ; les déplacements du robot sont déterministes. A chaque étape du processus, le robot analyse une image de chaque cible et obtient alors une observation de la nature de la cible : les deux cibles peuvent sembler être A (oAA), ou bien seulement la cible 1 (oAB), ou seulement la cible 2 (oBA), ou alors aucune des deux (oBB).

Dans le cadre probabiliste, la probabilité de recevoir une bonne observation de la cible $i \in \{1,2\}$, n'est pas vraiment connue, mais est approchée par $Pr(good_i \mid x,y) = \frac{1}{2}\left[1+\exp\left(-\frac{\sqrt{(x-x_i)^2+(y-y_i)^2}}{D}\right)\right]$ où $(x,y) = s_v \in \{1,\ldots,g\}^2$ est la position du robot, (x_i,y_i) la position de la cible i, et $D > 0$ une constante de normalisation. Les processus d'observation de chaque cible sont considérés indépendants. Alors, par exemple, $Pr(oAB \mid (x,y), A_1)$ est égal à $Pr(good_1 \mid (x,y)) \cdot Pr(good_2 \mid (x,y))$, $Pr(oAA \mid (x,y), A_1)$ à $Pr(good_1 \mid (x,y)) \cdot [1 - Pr(good_2 \mid (x,y))]$, etc. Chaque étape du processus avant d'atteindre une cible coûte 1, atteindre la cible A et y rester est récompensé par 100, et par -100 pour B. La stratégie provenant du modèle probabiliste a été calculée en tenant compte de l'Observabilité Mixte du problème, avec *APPL* [52], en utilisant une précision de 0.046 (la limite en mémoire est atteinte pour une précision supérieure) et $\gamma = 0.99$. Ce problème ne peut pas être résolu par l'algorithme exact pour PDMOM [2] puisque cela entraîne l'utilisation de toute la mémoire vive disponible après 15 itérations.

Avec la théorie des possibilités qualitatives, il est toujours possible d'observer correctement la cible : $\pi(good \mid x,y) = 1$. Ensuite, plus le robot est loin de la cible i, plus il est susceptible de mal l'observer (par exemple observer A au lieu de B), ce qui est une hypothèse raisonnable compte tenu du fait que le modèle d'observation est mal connu : $\pi(bad_i \mid x,y) = \frac{\sqrt{(x-x_i)^2+(y-y_i)^2}}{\sqrt{2}(g-1)}$. Ainsi, par exemple, $\pi(oAB \mid (x,y), A_1) = 1$, $\pi(oAA \mid (x,y), A_1) = \pi(bad_2 \mid x,y)$, $\pi(oBA \mid (x,y), A_1) = \min\{\pi(bad_1 \mid x,y), \pi(bad_2 \mid x,y)\}$, etc. Puisque la situation est complètement connue lorsque le robot est sur la position d'une cible (observation déterministe), il n'y a pas de risque d'être bloqué dans un état non satisfaisant, et c'est pourquoi le modèle π-PDMOM *optimiste* fonctionne bien. L'échelle $\mathcal{L}$ est composée de 0, 1, et toutes les valeurs possibles de $\pi(bad_i \mid x,y) \in [0,1]$. Notons qu'une construction de ce modèle avec une transformation probabilité-possibilité [30] aurait été équivalente. La distribution de préférence Ψ est égale à 0 pour tous les états du système, et à 1 pour les états $[(x_1,y_1), A_1]$ et $[(x_2,y_2), A_2]$ où (x_i,y_i) est la position de la cible i. Comme mentionné dans [62], la stratégie calculée garantit un plus court chemin vers les états buts : la stratégie tend à réduire le temps de la mission.

Les algorithmes pour π-PDMPO standards, qui n'exploitent pas l'observabilité mixte contrairement à notre modèle π-PDMOM, ne peuvent pas résoudre le problème même pour de très petites grilles 3×3. En effet, pour ce problème, $\#\mathcal{L} = 5$, $\#\mathcal{S}_v = 9$, et $\#\mathcal{S}_h = 2$. Ainsi, $\#\mathcal{S} = \#\mathcal{S}_v \cdot \#\mathcal{S}_h = 18$ et le nombre d'états de croyance est alors $\#\Pi_{\mathcal{L}}^{\mathcal{S}} = \mathcal{L}^{\#\mathcal{S}} - (\mathcal{L}-1)^{\#\mathcal{S}} = 5^{18} - 4^{18} \geqslant 3.7.10^{12}$ au lieu de 81 états avec un π-PDMOM. Par conséquent, les résultats expérimentaux qui suivent n'auraient pas pu être obtenus avec des π-PDMPO standards, ce qui justifie donc ce travail sur les π-PDMOM.

La comparaison des performances des modèles probabilistes et possibilistes peut se faire à l'aide des espérances de la somme des récompenses de leurs stratégies respectives : puisque la situation est complètement connue lorsque le robot est à la position d'une des cibles, il ne peut pas terminer en choisissant la cible B. Si k est le nombre d'étapes du processus pour identifier et atteindre la bonne cible, alors la somme des récompenses est $100 - k$.

Considérons maintenant qu'en réalité (donc ici pour les simulations), et contrairement à ce qui est décrit par le modèle, la situation en pratique fait que l'algorithme de vision artificielle utilisé par le robot est trompeur lorsque le robot est loin des cibles, *i.e.* si $\forall i \in \{1,2\}$, $\sqrt{(x-x_i)^2+(y-y_i)^2} > C$, avec C une constante positive, alors $Pr(good_i \mid x,y) = 1 - P_{bad} <$

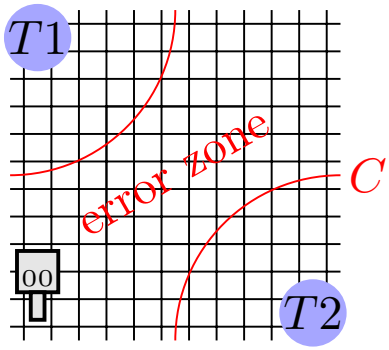


FIGURE II.2 – *Mission robotique de reconnaissance de cibles : deux cibles, la cible* 1 *(T1) et la cible* 2 *(T2) sont disposées sur une grille* $g \times g$*. Une des deux cibles est l'objet d'intérêt* A*, (et par élimination, l'autre cible est* B*) : soit* $T1 = A$*, soit* $T2 = A$*. Le robot reçoit des observations sur la nature de chaque cible : "*$Ti = A$*" ou "*$Ti = B$*" pour* $i \in \{1, 2\}$*, de manière bruitée. Plus le robot est proche d'une cible, moins l'observation à propos de la cible en question a de chances d'être fausse. Le robot doit reconnaître la cible* A *et l'atteindre. Cependant, sans avoir pu être pris en compte dans le modèle, lorsque le robot est dans une zone d'erreur (error zone en rouge) la probabilité qu'il observe mal* P_{bad} *est supérieure à* 0.5

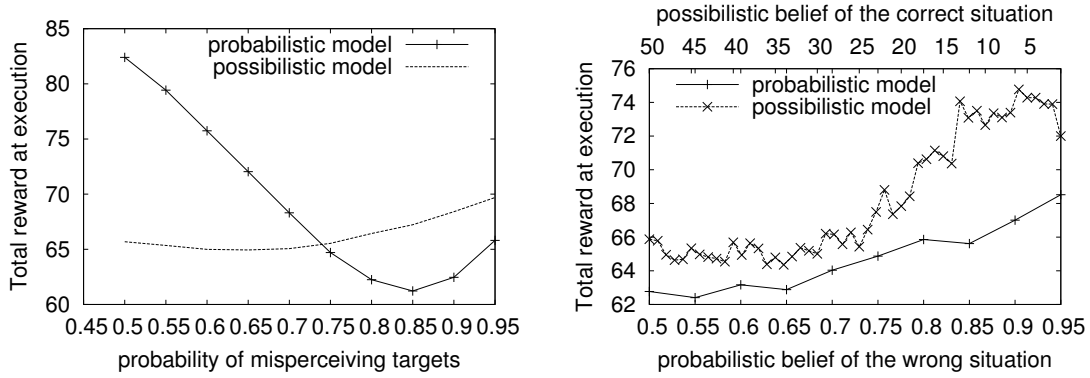
$\frac{1}{2}$. Dans tous les autres cas, le modèle probabiliste est bien décrit, *i.e.* en accord avec la réalité. La figure II.2 résume le problème, et indique la zone où le robot a une mauvaise perception par la dénomination "error zone". Pour les expérimentations numériques qui suivent, le nombre de simulations était de 10^4 pour calculer la moyenne de la somme des récompenses à l'exécution. La taille de la grille était de 10×10, $D = 10$ et $C = 4$.

La figure II.3.a montre que le modèle probabiliste est plus affecté par l'erreur introduite que le modèle possibiliste : elle représente la moyenne de la somme des récompenses à l'exécution obtenue par chaque modèle, comme une fonction de P_{bad}, la probabilité de mal observer une cible lorsque la position du robot est telle que $\sqrt{(x - x_i)^2 + (y - x_i)^2} > C$. Cela est dû au fait que la mise à jour possibiliste de l'état de croyance ne tient pas compte des nouvelles observations lorsque le robot en a déjà obtenu une plus fiable. Au contraire, le modèle probabiliste modifie l'état de croyance courant à chaque étape. En effet, puisqu'il n'y a que deux états cachés s_h^1 et s_h^2, si $\beta_h(s_h^1) < 1$, alors la normalisation possibiliste implique que $\beta_h(s_h^2) = 1$. La définition de la possibilité jointe de l'état et de l'observation (le minimum entre la distribution de possibilité sur l'état du système, *i.e.* l'état de croyance, et la possibilité de l'observation) assure que la possibilité jointe de s_h^1 et de l'observation obtenue, est plus petite que $\beta_h(s_h^1)$. L'équivalent possibiliste de l'équation de mise à jour de l'état de croyance (3) assure donc que l'état de croyance suivant se retrouve dans un des trois cas suivant :

- il est encore plus sceptique à propos de s_h^1 si l'observation est plus fiable, et confirme l'état de croyance précédent ($\pi\left(o_h \mid s_v, s_h^1, a\right)$ est plus petit que $\beta_h(s_h^1)$) ;
- il devient l'état de croyance opposé si l'observation est plus fiable et contredit l'état de croyance précédent ($\pi\left(o_h \mid s_v, s_h^2, a\right)$ est à la fois plus petit que $\beta_h(s_h^1)$ et que $\pi\left(o_h \mid s_v, s_h^1, a\right)$) ;
- il reste simplement le même si l'observation n'est pas plus informative que l'état de croyance courant.

Le théorème qui suit donne des conditions suffisantes menant à une mise à jour informative de l'état de croyance possibiliste. Classiquement un état de croyance $\beta_1 \in \Pi_{\mathcal{L}}^{\mathcal{S}}$ est dit plus spécifique qu'un état de croyance $\beta_2 \in \Pi_{\mathcal{L}}^{\mathcal{S}}$ si pour chaque $s \in \mathcal{S}$, $\beta_1(s) \leqslant \beta_2(s)$. La relation d'ordre $\preceq$ sur $\Pi_{\mathcal{L}}^{\mathcal{S}}$ peut alors être définie pour classer les états de croyance selon leur spécificité :

$$\beta_1 \preceq \beta_2 \Leftrightarrow \sum_{s \in \mathcal{S}} \beta_1(s) \leqslant \sum_{s \in \mathcal{S}} \beta_2(s)$$



(a) P_{bad} varie (b) β_0 varie, $P_{bad} = 0.8$

FIGURE II.3 – *Comparaison des moyennes de la somme des récompenses à l'exécution, pour les modèles probabilistes et possibilistes.*

Notons que si β_1 est plus spécifique que β_2, alors $\beta_1 \preceq \beta_2$.

Théoreme 7

Soit $\beta_0 \in \Pi_{\mathcal{L}}^{\mathcal{S}}$ l'état de croyance initial modélisant l'ignorance totale, i.e. pour tous les $s \in \mathcal{S}$, $\beta_0(s) = 1$. Si la fonction de transition est déterministe, et si les observations ne sont pas informatives, i.e. $\forall s' \in \mathcal{S}$, $\forall a \in \mathcal{A}$, $\forall o' \in \mathcal{O}$, $\pi(o' \mid s', a) = 1$, alors si l'état de croyance $\beta_{t+1} \in \Pi_{\mathcal{L}}^{\mathcal{S}}$ est le résultat de la mise à jour de l'état de croyance $\beta_t \in \Pi_{\mathcal{L}}^{\mathcal{S}}$, $\beta_{t+1} \preceq \beta_t$. Ce résultat reste valable si pour chaque action la fonction de transition est l'identité et $\forall (o', \tilde{o}) \in \mathcal{O}^2$, $\forall (a, \overline{a}) \in \mathcal{A}^2$ et $\forall s' \neq \tilde{s} \in \mathcal{S}$ t.q. $\pi(o' \mid s', a) < 1_{\mathcal{L}}$, $\pi(o' \mid s', a) \neq \pi(\tilde{o} \mid \tilde{s}, \overline{a})$.

La mise à jour probabiliste quant à elle ne permet pas à l'état de croyance de devenir directement l'état de croyance opposé, ou d'ignorer les observations moins fiables : le robot se dirige d'abord vers la mauvaise cible car il est initialement trop loin des deux cibles et les observe mal. Lorsqu'il est proche de cette cible, il reçoit de bonnes observations, et change petit à petit d'état de croyance : ce dernier devient assez informatif pour le convaincre de se diriger vers la cible A. Cependant, il passe alors inévitablement par la zone d'erreur : cela modifie peu à peu son état de croyance, qui devient faux avec grande probabilité, et le robot se retrouve dans la situation initiale. Il perd ainsi beaucoup de temps à sortir de cette boucle. On peut voir que la moyenne de la somme des récompenses croît lorsque la probabilité de mal observer, P_{bad}, est très grande : cela s'explique par le fait que cette grande erreur mène le robot à atteindre la mauvaise cible plus rapidement, et donc à être quasiment sûr que la cible A est l'autre cible.

Maintenant, fixons $P_{bad} = 0.8$ et évaluons la moyenne de la somme des récompenses à l'exécution pour différents faux états de croyance initiaux : la figure II.3.b illustre cette évaluation, avec les mêmes paramètres que pour la précédente expérimentation : nous comparons ici le modèle possibiliste, et la probabiliste lorsque l'état de croyance initial est fortement orientée vers la mauvaise cible (*i.e.* l'agent pense fortement que la cible 1 est B alors que c'est A en réalité). Notons que l'état de croyance possibiliste en la bonne cible décroît lorsque la nécessité en la mauvaise croît. Cette figure montre que le modèle possibiliste mène à de meilleures récompenses à l'exécution si l'état de croyance initial est mauvais et la fonction d'observation est imprécise : notons cependant que pour $\mathcal{P}_{bad} \leqslant 0.6$, la politique probabiliste est plus efficace [1].

1. L'implémentation de l'algorithme de résolution, ainsi que la description de ce problème de reconnaissance qui en est l'entrée, sont disponibles sur le dépôt accessible à l'adresse `https://github.com/drougui/ppudd` : le problème peut être simulé en utilisant la stratégie optimale possibiliste calculée par l'algorithme.

II.4 Conclusion

Nous avons proposé un algorithme d'Itération sur les Valeurs (IV) pour les PDM possibilistes. Celui-ci calcule une stratégie optimale stationnaire pour un horizon indéterminé contrairement aux méthodes précédentes. Une preuve complète de la convergence a été fournie : elle repose sur l'existence d'une action "rester" intermédiaire. Celle-ci est utilisée uniquement pour maintenir le processus dans les états buts. Enfin, le nouveau modèle des PDM possibilistes qualitatifs à observabilité mixte, a été présenté, et sa complexité est exponentiellement plus petite que celle des PDMPO possibilistes qualitatifs. De ce fait, nous avons pu comparer les π-PDMOM avec leurs équivalents probabilistes sur un problème robotique réaliste. Nos résultats expérimentaux montrent que ces stratégies possibilistes peuvent être plus performantes que les stratégies issues du modèle probabiliste lorsque la fonction d'observation n'est pas connue précisément.

La version de l'algorithme d'Itération sur les Valeurs pour π-PDM pessimiste peut aussi être construite, mais l'optimalité de la stratégie renvoyée semble dure à prouver. Les travaux [77] et [59] peuvent être utiles pour obtenir des résultats à propos de ces π-PDM pessimistes, afin de résoudre des problèmes contenant des situations dangereuses.

Finalement, comme cela a été mis en évidence par les expériences, bien que les π-PDMPO soient basés sur un modèle d'incertitude plus simple en termes de complexité que les PDMPO probabilistes, le processus de croyance peut avoir un comportement intéressant. Pour certaines conditions suffisantes données par le théorème 7, l'état de croyance n'est pas modifié par des informations moins fiables que celles accumulées avant elles, mais est capable de se transformer en un état de croyance quasi opposé si une information qui le suggère et qui est plus fiable est reçue. Des problèmes plus complexes doivent être étudiés pour avoir une meilleure idée de son comportement dans un panel de situations plus important. Cependant, les π-PDMPO avec un espace d'état trop grand (ou π-PDMOM avec un trop grand espace $\mathcal{S}_h$) ne peuvent pas être résolus en temps raisonnables par les algorithmes développés jusqu'à aujourd'hui. Le chapitre suivant présente et utilise d'autres structures du problème, décrites par l'homologue possibiliste qualitatif du modèle des *PDMPO factorisés* : ces structures mènent à des calculs de stratégies possibilistes plus simples, et permettent de résoudre de nombreux problèmes de planification.

III Développement d'Algorithmes Symboliques pour la Résolution des π-PDMPO

Dans ce chapitre, nous proposons l'étude des modèles π-PDMOM dans le but de résoudre de très grands problèmes de planification lorsqu'ils sont structurés. Inspirés par l'algorithme *Stochastic Planning Using Decision Diagrams* (*SPUDD*) construit pour résoudre les PDM probabilistes factorisés, nous avons mis en place un algorithme symbolique appelé PPUDD conçu pour résoudre les π-PDMOM. Tandis que le nombre de feuilles des arbres de décision utilisés par *SPUDD* peuvent devenir aussi grand que la taille de l'espace des états puisque leurs valeurs sont des nombres réels agrégés par des additions et des multiplications, le nombre de feuilles de *PPUDD* est borné par le nombre d'éléments dans $\mathcal{L}$ car leurs valeurs restent dans l'échelle finie $\mathcal{L}$ via les opérations min et max seulement. Enfin, nous présentons un π-PDMOM satisfaisant certaines hypothèses d'indépendance sur les variables visibles, cachées, et d'observation. Ce dernier résulte en un π-PDM factorisé, sur lequel *PPUDD* peut être lancé. Nos résultats expérimentaux montrent que le temps de calcul de *PPUDD* est beaucoup plus petit que *Symbolic-HSVI* et *APPL* pour les versions possibilistes et probabilistes du même benchmark, tout en fournissant des stratégies de bonne qualité. Les performances des stratégies calculées par *PPUDD* ont été testées pendant la compétition internationale de planification probabiliste (*IPPC* 2014) dont les résultats sont exposés ici.

III.1 Introduction

Les travaux sur les π-PDM(MO) présentés précédemment ne tirent pas totalement avantage de la structure du problème, *i.e.* les parties visibles ou cachées de l'état peuvent être elles-même factorisées en plusieurs variables d'états. Dans le cadre probabiliste, les PDM factorisés et les méthodes de calcul symboliques [11, 36] ont été étudiés intensivement dans le but de raisonner directement au niveau des variables plutôt que sur l'espace d'état. Un travail récent sur ces problématiques est par exemple [60]. Le célèbre algorithme *SPUDD* [36] résout des PDM factorisés en utilisant des représentations symboliques de la fonction valeur et des stratégies sous la forme d'arbres de décision algébriques (*ADDs*) [3], qui représentent de manière compacte les fonctions réelles de variables booléennes : les *ADDs* sont des arbres dont les noeuds représentent les variables d'état et les feuilles sont les valeurs de la fonction. Au lieu de mettre à jour les valeurs de la fonction pour chaque état individuellement à chaque itération de l'algorithme, ils sont aggrégés dans les *ADDs* et les opérations sont effectuées de manière symbolique directement entre les *ADDs* sur plusieurs variables en même temps. Cependant, *SPUDD* est limité par la manipulation potentielle d'énormes *ADDs* dans le pire des cas : par exemple, l'espérance implique des additions et des multiplications sur des valeurs

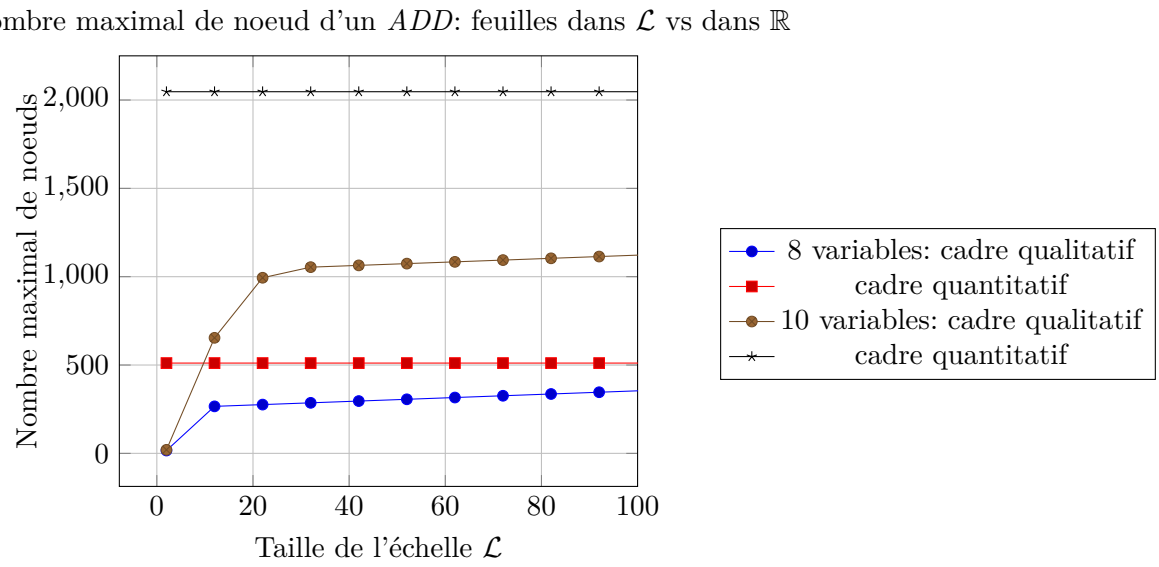


FIGURE III.1 – *La taille maximale (nombre total de noeuds) d'un ADD dont les valeurs sont dans $\mathcal{L}$ est limité : cette taille maximale est représentée par les courbes avec les ronds bleus et marrons, en fonction de la taille de $\mathcal{L}$. Lorsque les feuilles des ADDs sont dans $\mathbb{R}$, le nombre de ses noeuds est potentiellement exponentiel en le nombre de variables : la borne supérieure est représentée par les courbes avec les carrés rouges et noirs (fonctions constantes de la taille de $\mathcal{L}$).*

réelles (probabilités et récompenses), créant de nouvelles valeurs entres elles, de manière à ce que le nombre de feuille des *ADDs* puisse devenir égal à l'espace d'état, *i.e.* exponentiel en le nombre de variables d'état.

Ainsi, le travail présenté ici est motivé par la simple observation que les **opérations symboliques avec des PDM possibilistes devraient nécessairement limiter la taille des *ADDs*** : en effet, ce formalisme opère sur une échelle *finie* $\mathcal{L}$ avec seulement les opérations max et min, ce qui implique que les valeurs manipulées restent dans l'échelle finie $\mathcal{L}$, qui est généralement beaucoup plus petite que le nombre d'états.

La figure III.1 montre que les *ADDs* utilisés dans le cadre possibiliste a un nombre limité de noeuds puisque le nombre de feuilles est au plus égal à la taille de l'échelle possibiliste qualitative $\mathcal{L}$: la taille maximale (nombre maximal de noeuds) d'un *ADD* dont les feuilles sont dans $\mathcal{L}$, est représenté comme une fonction de $\#\mathcal{L}$, dans le cas de 8 et 10 variables.

Dans ce chapitre, nous présentons un algorithme basé sur la programmation dynamique symbolique pour résoudre les π-PDMOM factorisés appelé Possibilistic Planning Using Decision Diagram (*PPUDD*). Cette contribution seule n'est pas suffisante puisque les variables de croyance ont un nombre de valeurs exponentiel en la taille de l'espace des états cachés. Donc, notre seconde contribution est un théorème visant à factoriser l'état de croyance en de nombreuses variables de croyance marginales lorsque certaines hypothèses d'indépendance sur les variables d'état et d'observation d'un π-PDMOM sont vérifiées : cela permet de résoudre certains problèmes dont les calculs sont inabordable. Notons que notre idée de factorisation de l'état de croyance est assez général pour être valable pour les modèles probabilistes. Enfin, les performances de *PPUDD* sont comparées à celles de *symbolic HSVI* [69] (une version symbolique de l'algorithme pour PDMPO appelé *HSVI* [71]) et *APPL* [43, 52] (déjà utilisé dans le chapitre précédent, et basé sur *SARSOP*) sous observabilité mixte. Les résultats obtenus étant prometteurs, nous avons participé à la compétition internationale de planification probabiliste (*IPPC* 2014) et les résultats de *PPUDD* durant la session entièrement observable d'*IPPC* 2014 sont présentés et discutés. Un algorithme dédié à la résolution des π-PDMOM en utilisant des *ADDs* est disponible dans le dépôt `https://github.com/drougui/ppudd`).

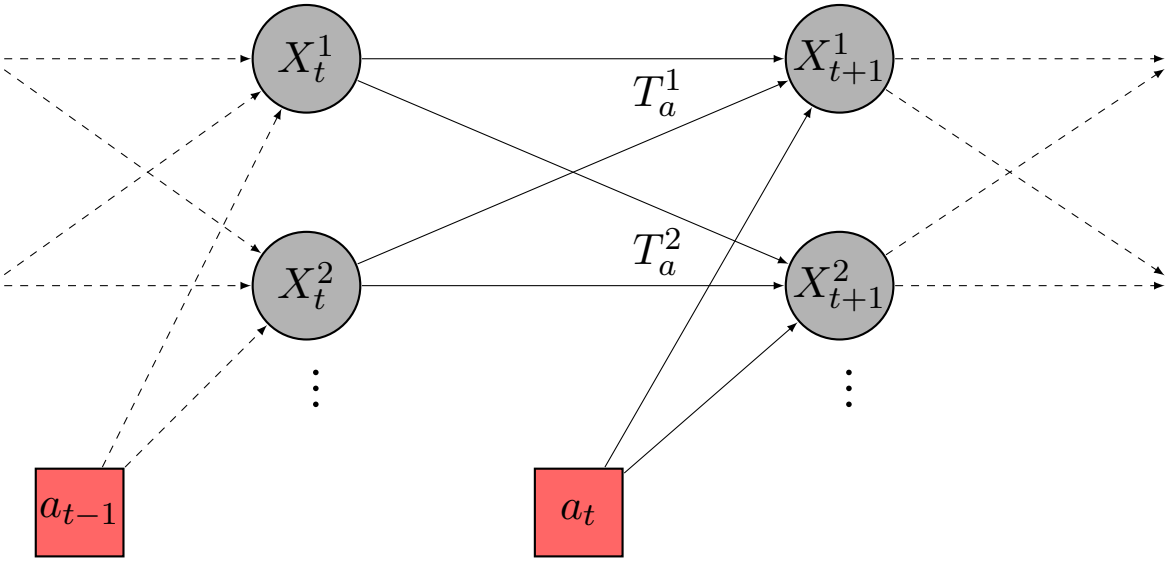


FIGURE III.2 – *Réseau dynamique bayésien pour des (π-)PDM : dans le cadre possibiliste (resp. probabiliste)* T_a^i *est la distribution de possibilité (resp. probabilité) de transition sur la variable d'état* X_{t+1}^i *conditionnellement à l'action sélectionnée* $a \in \mathcal{A}$ *et à ses parents* $parents(X_{t+1}^i) \subseteq \{X_t^1, \dots, X_t^n\}$ *(i.e.* $parents(X_{t+1}^i)$ *est un sous ensemble de l'ensemble des variables d'état courantes) où* $n \geqslant 1$ *est le nombre de variables décrivant l'espace d'état.*

III.2 Résoudre des π-PDMOM par la Programmation Dynamique Symbolique

Les PDM factorisés [36] ont été utilisés pour résoudre plus rapidement les problèmes structurés de décision séquentielle, sous incertitude probabiliste, en raisonnant symboliquement sur les fonctions des états du systèmes, à travers des arbres de décision algébriques. Inspiré par ce travail, cette section présente la résolution symbolique des π-PDMOM factorisés : dans ce modèle, l'espace des états visibles $\mathcal{S}_v$, l'espace des états cachés $\mathcal{S}_h$ et l'ensemble des observations $\mathcal{O}_h$ sont tels que l'espace d'état du π-PDM résultant (basé sur l'espace des états de croyances et l'espace des variables visibles) est sous la forme $\mathcal{S}_v^1 \times \cdots \times \mathcal{S}_v^m \times \Pi_{\mathcal{L}}^{\mathcal{S}_h}$, où chacun de ces espaces sont finis. Nous verrons dans la section suivante comment $\Pi_{\mathcal{L}}^{\mathcal{S}_h}$ peut être factorisé de cette manière grâce à la factorisation de $\mathcal{S}_h$ et $\mathcal{O}_h$. La factorisation de la variable de la croyance probabiliste dans [12, 67] est approximative, tandis que celle présentée ici est exacte. Puisque les espaces finis de taille K peuvent être eux-même factorisés en $\lceil \log_2 K \rceil$ espaces binaires [36], nous pouvons faire l'hypothèse que nous raisonnons sur un π-PDM dont l'espace d'état est noté $\mathcal{X}$ et entièrement décrit par les variables $(X^1, \dots, X^n)$, avec $n \in \mathbb{N}^*$ et $\forall i$, $X^i \in \{\top, \bot\}$: $\mathcal{X} = \{\top, \bot\}^n$.

Rappelons que les réseaux bayésiens dynamiques (*DBNs*) [21] déjà utilisés en section I.4 (dans le diagramme d'influence figure I.4) et dans le chapitre précédent (figure II.1 illustrant la structure d'observabilité mixte) sont des représentations graphique très utiles pour les processus étudiés. Un *DBN* représentant la structure d'un π-PDM factorisé est dessiné dans la figure III.2 : les variables d'état à une étape de temps donnée $t \geqslant 0$ sont notées $X_t = (X_t^i)_{i=1}^n$ (variables courantes), et $(X_{t+1}^i)_{i=1}^n$ sont les variables d'état à l'étape de temps $t+1$ (variable suivante). Dans le cadre des *DBNs*, $parents(X_{t+1}^i)$ est l'ensemble des variables d'état dont la variable d'état suivante X_{t+1}^i dépend, *i.e.* une variable Y, représentée par un noeud dans le *DBN*, est dans $parents(X_{t+1}^i)$ si et seulement si il y a une flèche de Y à X_{t+1}^j. Nous supposons que $parents(X_{t+1}^i) \subseteq \{X_t^1, \dots, X_t^n\}$, *i.e.* les parents de la variable d'état suivante X_{t+1}^i font

partie des variables d'état courantes $\{ X_t^1, \ldots, X_t^n \}$: il ne peut y avoir de flèches entre les variables d'état de la même étape de temps.

Avec les notations du π-PDMOM, les hypothèses du réseau bayésien de la figure III.2 nous permettent de calculer la distribution de possibilité jointe : $\pi(s'_v, \beta'_h \mid s_v, \beta_h, a) = \pi(X' \mid X, a) = \min_{i=1}^{n} \pi(X'_i \mid parents(X'_i), a)$, où, étant donné l'étape de temps t, les variables primées sont les variables concernant l'étape de temps $t+1$ (variables suivantes), et les variables non-primées sont les variables courantes (à l'étape de temps t) : par exemple, X'_i est la notation pour X_{t+1}^i, et X_i celle pour X_t^i. Ainsi, un π-PDMOM factorisé peut être défini par des fonctions de transition $T_a^i = \pi(X'_i \mid parents(X'_i), a)$ pour chaque action a et chaque variable X'_i (si les transitions sont stationnaires).

Chaque fonction de transition peut être représentée par un arbre de décision algébrique (*ADD*) [3]. Un *ADD*, comme illustré dans la figure III.3a, est un arbre représentant de manière compacte une fonction réelle de variables binaires, dont les sous-graphes identiques sont confondus et les feuilles valant zéro ne sont pas mémorisées. Les notations suivantes sont utilisées pour rendre explicite le fait que nous travaillons avec des fonctions symboliques représentées par des *ADDs* :

— $\boxed{\min}\{f, g\}$ où f et g sont deux *ADDs* ;

— $\boxed{\max}_{X_i} f = \boxed{\max}\left\{ f^{X_i=0}, f^{X_i=1} \right\}$,

qui peut être facilement calculé car les *ADDs* sont construits sur la base de l'expension de Shanon : $f = \overline{X_i} \cdot f^{X_i=0} + X_i \cdot f^{X_i=1}$ où $f^{X_i=1}$ et $f^{X_i=0}$ sont les sous graphes représentant les cofacteurs de Shanon positifs et négatifs (*cf.* figure III.3a).

Le schéma de programmation dynamique, *i.e.* la ligne 8 de l'algorithme d'itération sur les valeurs 3 du chapitre précédent, peut être réécrite sous forme symbolique, de telle sorte que les états soient globalement mis à jour en un coup, plutôt qu'individuellement : en notant $X = (X_1, \ldots, X_n)$ la variable d'état courante et $X' = (X'_1, \ldots, X'_n)$ la suivante, la Q-valeur pour une action $a \in \mathcal{A}$ est $\overline{q^a} = \overline{q^a}(X) = \boxed{\max}_{X'} \boxed{\min}\left\{ \pi(X' \mid X, a), \overline{U^*}(X') \right\}$. Le calcul de cet *ADD* ($\overline{q^a}$) peut être décomposé en plusieurs calculs en utilisant la proposition suivante :

Propriété III.2.1 (*Régression possibiliste de la fonction valeur*)

Considérons la fonction valeur courante $\overline{U^*} : \{\top, \bot\}^n \to \mathcal{L}$. *Pour une action donnée* $a \in \mathcal{A}$, *définissons :*

— $\overline{q_0^a} = \overline{U^*}(X'_1, \cdots, X'_n)$,

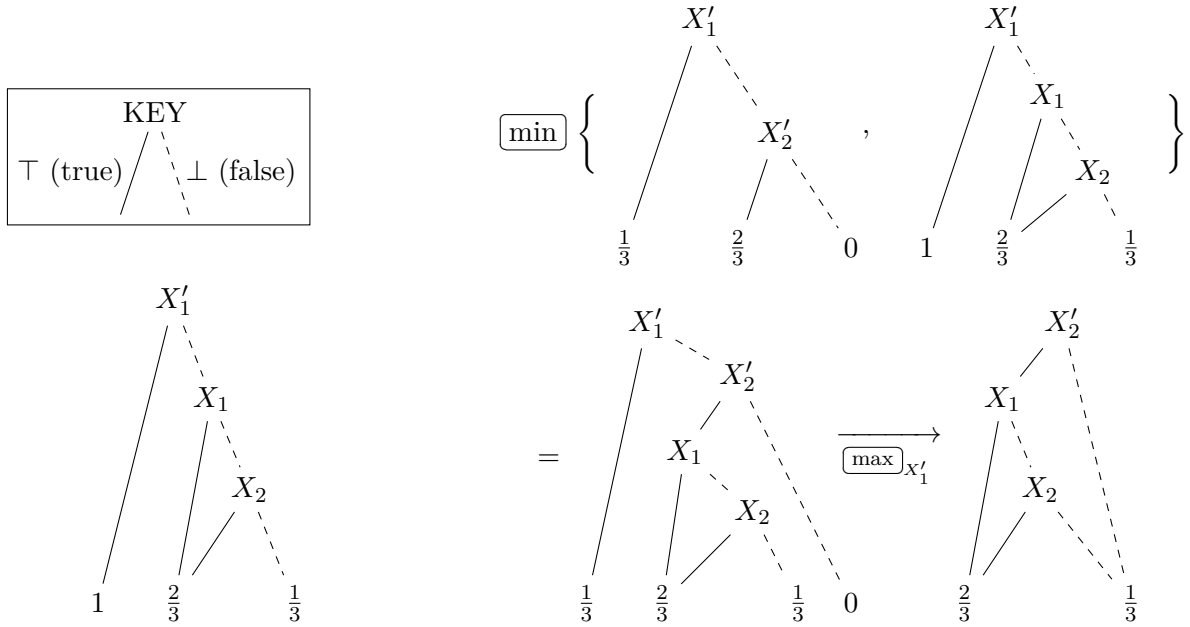
— $\overline{q_i^a} = \max_{X'_i \in \{\top, \bot\}} \min\left\{ \pi(X'_i \mid parents(X'_i), a), \overline{q_{i-1}^a} \right\}$.

Alors, la Q-valeur possibiliste d'une action a *est* $\overline{q^a} = \overline{q_n^a}$, *qui dépend des variables* $X_1, \ldots, X_n$, *et la fonction valeur suivante est* $\overline{U^*}(X_1, \ldots, X_n) = \max_{a \in \mathcal{A}} \overline{q_n^a}(X_1, \ldots, X_n)$.

La Q-valeur de l'action a, représentée par un *ADD*, peut être alors calculée en plusieurs étapes, une pour chaque variable d'état suivante $X'_i, 1 \leqslant i \leqslant n$. La figure III.3b illustre les calculs possibilistes effectués entre les arbres de décision algébriques (*ADDs*) pour calculer la Q-valeur d'une action.

L'algorithme 4 est une version symbolique de l'algorithme d'itération sur les valeurs pour π-PDM dans le chapitre précédent, algorithme 3), qui utilise le schéma de régression défini dans la proposition III.2.1. Inspiré de *SPUDD* [36], *PPUDD* signifie *Possibilistic Planning Using Decision Diagrams*. Comme pour *SPUDD*, l'action de transformer les variables non primées en variables primées est nécessaire à chaque itération (*cf.* ligne 5 de l'algorithme 4 et figure III.3b). Cette opération sert à différentier les états suivants des états courants lors des opérations entre les ADDs. Les lignes 4-9 appliquent la proposition III.2.1 et correspondent à la ligne 8 de l'algorithme 3.

Nous avons mentionné au début de la section que l'espace des états de croyance $\Pi_{\mathcal{L}}^{\mathcal{S}_h}$ pourrait être décrit par $\lceil \log_2 K \rceil$ variables binaires où $K = \#\mathcal{L}^{\#\mathcal{S}_h} - (\#\mathcal{L} - 1)^{\#\mathcal{S}_h}$. Cependant,



(a) *ADD représentant T_1^a de la figure III.2*

(b) *Calcul symbolique de la Q-valeur courante sous forme d'un ADD : elle est combinée à la fonction de transition représentée par un ADD dans la figure III.3a*

FIGURE III.3 – *Exemple d'arbre de décision algébrique comme utilisé dans l'algorithme PPUDD*

Algorithm 4: PPUDD (calcul pour un horizon indéterminé)

$\overline{U^*} \leftarrow 0$; $\overline{U^c} \leftarrow \Psi$; $\overline{\delta} \leftarrow \widehat{a}$;
while $\overline{U^*} \neq \overline{U^c}$ **do**
  $\overline{U^*} \leftarrow \overline{U^c}$;
  for $a \in \mathcal{A}$ **do**
    $\overline{q^a} \leftarrow$ swap each X_i variable in $\overline{U^*}$ with X_i' ;
    for $1 \leqslant i \leqslant n$ **do**
      $\overline{q^a} \leftarrow \boxed{\min}\left\{\overline{q^a}, \pi\left(X_i' \,\middle|\, parents(X_i'), a\right)\right\}$;
      $\overline{q^a} \leftarrow \boxed{\max}_{X_i'} \overline{q^a}$;
    $\overline{U^c} \leftarrow \boxed{\max}\left\{\overline{q^a}, \overline{U^c}\right\}$;
    update $\overline{\delta}$ to a where $\overline{q^a} = \overline{U^c}$ and $\overline{U^c} > \overline{U^*}$;
return $\overline{U^*}$, $\overline{\delta^*}$;

ce K peut être très grand, ainsi nous proposons dans la section qui suit, une méthode pour exploiter la factorisation de $\mathcal{S}_h$ et $\mathcal{O}_h$ dans le but de factoriser $\Pi_{\mathcal{L}}^{\mathcal{S}_h}$ en plusieurs variables, ce qui décomposera les *ADDs* de transition en plus petits *ADDs* facilement manipulables. Notons que *PPUDD* peut résoudre les π-PDMOM même si cette factorisation de la croyance n'est pas possible, mais les *ADDs* manipulés sont plus gros dans ce cas.

Le chapitre précédent met en évidence qu'un prétraitement est nécessaire pour traduire un π-PDMOM en un π-PDM dont l'espace d'état est $\mathcal{X}$. Nous pouvons alors raisonner sur l'espace d'état entièrement visible par l'agent $\mathcal{X} = S_v \times \Pi_{\mathcal{L}}^{\mathcal{S}_h}$ et résoudre le π-PDMOM en tant que π-PDM. La section suivante fait le lien entre les propriétés structurelles d'un π-PDMOM, concernant les dépendences des variables originales (visibles, cachées et d'observation), à la factorisation du problème traité *i.e.* du π-PDM résultant, défini sur l'espace d'état $S_v \times \Pi_{\mathcal{L}}^{\mathcal{S}_h}$: la factorisation résultante concerne alors les dépendances des variables visibles et des variables de croyance.

III.3 Factorisation de la Variable de Croyance d'un π-PDMOM

La factorisation de la variable de croyance requiert trois hypothèse d'indépendance sur les variables du π-PDMOM, qui sont illustrées à travers le problème classique *Rocksample* [71].

III.3.1 Exemple de Motivation

Un robot se déplaçant sur une grille $g \times g$ doit collecter des échantillons scientifiques à partir de pierres dites intéressantes parmi R pierres, et enfin d'atteindre la sortie de la grille. Il connait les positions des R pierres $(x_i, y_i)_{i=1}^{R}$ mais pas lesquelles sont intéressantes. Les pierres intéressantes sont appelées les "bonnes" pierres. Cependant, échantillonner une pierre coûte cher : le robot est donc équipé d'un capteur qu'il peut utiliser pour déterminer si une pierre est "bonne" ou non ("mauvaise"). Lorsqu'une pierre est échantillonnée, elle devient (ou reste) "mauvaise" (plus intéressante scientifiquement). A la fin de la mission, le robot doit atteindre la position de sortie de la grille. Décrivons le PDMOM associé :

- $\mathcal{S}_v$ représente l'ensemble des positions possibles du robot en plus de la sortie ($\#\mathcal{S}_v = g^2 + 1$) ;
- $\mathcal{S}_h$ représente l'ensemble des natures possibles de chacune des pierres : $\mathcal{S}_h = \mathcal{S}_h^1 \times \ldots \times \mathcal{S}_h^R$, avec $\forall 1 \leqslant i \leqslant R$, $\mathcal{S}_h^i = \{\text{“}bonne''\text{,“}mauvaise''\}$;
- $\mathcal{A}$ contient les déplacements (déterministes) dans les 4 directions $(a_{north}, a_{east}, a_{south}, a_{west})$, tester la pierre numéro i, (a_{check_i}) $\forall 1 \leqslant i \leqslant R$, et échantillonner la pierre courante, (a_{sample}) ;
- $\mathcal{O} = \{o_{good}, o_{bad}\}$ sont les différentes réponses possibles des capteurs pour la pierre testée.

La dynamique des observations est la suivante : plus le robot est proche de la pierre testée, mieux il observe sa nature. Dans le problème probabiliste original, la probabilité d'une observation correcte est égale à $\frac{1}{2}\left(1 + e^{-c\sqrt{(x_r - x_i)^2 + (y_r - y_i)^2}}\right)$ avec $c > 0$, une constante (plus c est petit, plus le capteur est efficace). Le robot reçoit la récompense $+10$ (resp. -10) pour chaque bonne (resp. mauvaise) pierre échantillonnée, et $+10$ lorsqu'il atteint la sortie de la grille.

Dans cadre possibiliste, la fonction d'observation est approximée en utilisant une distance critique au-delà de laquelle tester une pierre n'est pas informatif : $\pi(o_i' \mid s_i', a, s_v) = 1\ \forall o_i' \in \mathcal{O}_i$. Le degré de possibilité d'une observation erronée devient zéro lorsque le robot se trouve sur la pierre testée, et devient le plus petit degré de possibilité non nul sinon. Enfin, puisque la sémantique possibiliste ne permet pas de sommer les récompenses, une variable additionnelle

$s_v^2 \in \{1, \ldots, R\}$ est introduite : elle compte le nombre de pierres testées. La préférence qualitative de l'échantillonage est définie par $\Psi(s) = \frac{R+2-s_v^2}{R+2} \in \mathcal{L}$ si la position est la sortie, et zéro sinon. Enfin, la position du robot est notée $s_v^1 \in \mathcal{S}_v^1$ et donc la variable d'état visible est finalement $s_v = (s_v^1, s_v^2) \in \mathcal{S}_v^1 \times \mathcal{S}_v^2 = \mathcal{S}_v$.

Les observations $\{o_{good}, o_{bad}\}$ pour la pierre testée peuvent être modélisées de manière équivalente comme le produit cartésien des observations $\{o_{good_1}, o_{bad_1}\} \times \cdots \times \{o_{good_R}, o_{bad_R}\}$ pour chaque pierre. En utilisant cette modélisation, les espaces d'états et d'observations sont respectivement factorisés de la manière suivante, $\mathcal{S}_v^1 \times \ldots \times \mathcal{S}_v^m \times \mathcal{S}_h^1 \times \ldots \times \mathcal{S}_h^l$, et $\mathcal{O} = \mathcal{O}^1 \times \ldots \times \mathcal{O}^l$, et nous pouvons maintenant associer une variable d'observation $O^j \in \mathcal{O}^j$ à la variable d'état cachée correspondante $S_h^j \in \mathcal{S}_h^j$. Cela permet de raisonner sur le *DBN* de la figure III.4, qui exprime trois hypothèses importantes qui nous permettrons de factoriser l'état de croyance :

1. toutes les variables d'état $S_v^1, S_v^2, \ldots, S_v^m, S_h^1, S_h^2, \ldots, S_h^l$ sont indépendantes post-action, et une variable d'état suivante ne dépend par des variables d'état cachées courantes.
2. une variable d'état cachée ne dépend pas d'autres variables d'état cachées précédentes : par exemple la nature d'une pierre est indépendante de la nature des autres pierres.
3. une variable d'observation est disponible pour chaque variable d'état cachée, et dépend de cet état. Elle ne dépend pas d'autres variables d'état cachées, où des variables visibles courantes, mais des variables visibles précédentes et de l'action choisie :

Chaque variable d'observation est en effet seulement associée à la nature de la pierre correspondante. La qualité de l'observation dépend de la position du robot *i.e.* une variable d'état visible courante, ce qui n'est pas autorisé par le *DBN* : heureusement, puisque les déplacements sont déterministes, nous contournons le problème en considérant que cette qualité dépend de la position précédente et de l'action choisie, ce qui est équivalent ici.

III.3.2 Conséquences des Hypothèses de Factorisation

Dans cette section, nous présentons le résultat formel provenant des trois précédentes hypothèses : ce résultat est la factorisation de $\Pi_{\mathcal{L}}^{\mathcal{S}_h}$ en le produit cartésien $\bigtimes_{j=1}^{l} \Pi_{\mathcal{L}}^{\mathcal{S}_h^j}$. En effet, l'état de croyance β_h à propos de l'état caché du système $s_h \in \mathcal{S}_h$ peut être représenté des croyances marginales $\beta_h^j \in \Pi_{\mathcal{L}}^{\mathcal{S}_h^j}$ sur les états cachés $s^j \in \mathcal{S}_h^j$, $\forall j \in \{1, \ldots, l\}$.

Le *critère de d-Séparation* [74] qui permet de montrer graphiquement des résultats d'indépendance à partir du *DBN*, se cache derrière les résultats fournis. Comme expliqué en section III.2, un *DBN* peut être construit à partir de relations d'indépendances. Notons $X \perp\!\!\!\perp Y \mid Z$ l'assertion "X est indépendant de Y conditionnellement à Z" : rappelons que pour une définition donnée de la relation d'indépendance, par exemple l'indépendance probabiliste, ou l'indépendance possibiliste causale, le *DBN* est construit de telle sorte que pour chaque variable X, $X \perp\!\!\!\perp \mathit{nondescend}(X) \mid \mathit{parents}(X)$, où $\mathit{nondescend}(X)$ est l'ensemble des variables qui ne sont pas des descendants de X. Si l'indépendance utilisée obéit aux axiomes des *semi-graphoides* [55, 75], le critère graphique appelé d-Séparation peut être utilisé pour identifier certaines indépendances sur le *DBN*. Ce critère est utilisé, par exemple, dans le cadre des probabilités dans le travail [78].

Tout d'abord, comme le *DBN* de la figure III.4 représente des hypothèses d'indépendance causales, la distribution de possibilité sur la variable d'état visible numéro i $s_{v,t+1}^i \in \mathcal{S}_v^i$ sachant les variables précédentes peut s'écrire :

$$\pi\left(s_{v,t+1}^i \,\middle|\, s_{v,t}, a_t\right) = \Pi\left(S_{v,t+1}^i = s_{v,t+1}^i \,\middle|\, S_{v,t} = s_{v,t}, a_t\right); \tag{III.1}$$

Définissons l'information i_t connue par l'agent à l'étape de temps $t \geqslant 1$ lorsque le modèle est un (π-)PDMOM : $i_0 = \{ s_{v,0} \}$, et pour chaque étape de temps $t \geqslant 1$, $i_t = \{ o_t, s_{v,t}, a_{t-1}, i_{t-1} \}$. La variable correspondante est notée I_t. Le théorème suivant assure que l'état de croyance courant peut être décomposé en états de croyance marginaux dépendant de l'information courante.

Théoreme 8 (*Indépendance des variables d'état cachées sachant i_t*)

Considérons un π-PDMOM décrit par le DBN de la figure III.4. Si les variables d'état cachées initiales $S^1_{h,0}, \ldots, S^l_{h,0}$ sont indépendantes, alors à chaque étape de temps $t > 0$ l'état de croyance sur les états cachés peut s'écrire

$$\beta_{h,t} = \min_{j=1}^{l} \beta^j_{h,t}$$

avec $\forall s \in \mathcal{S}^j_h$, $\beta^j_{h,t}(s) = \Pi\left(S^j_{h,t} = s \,\middle|\, I_t = i_t \right)$ l'état de croyance marginal concernant les états cachés du système de l'ensemble $\mathcal{S}^j_h$.

Grâce au théorème précédent, l'espace d'état visible par l'agent peut se réécrire $\mathcal{S}^1_v \times \ldots \times$

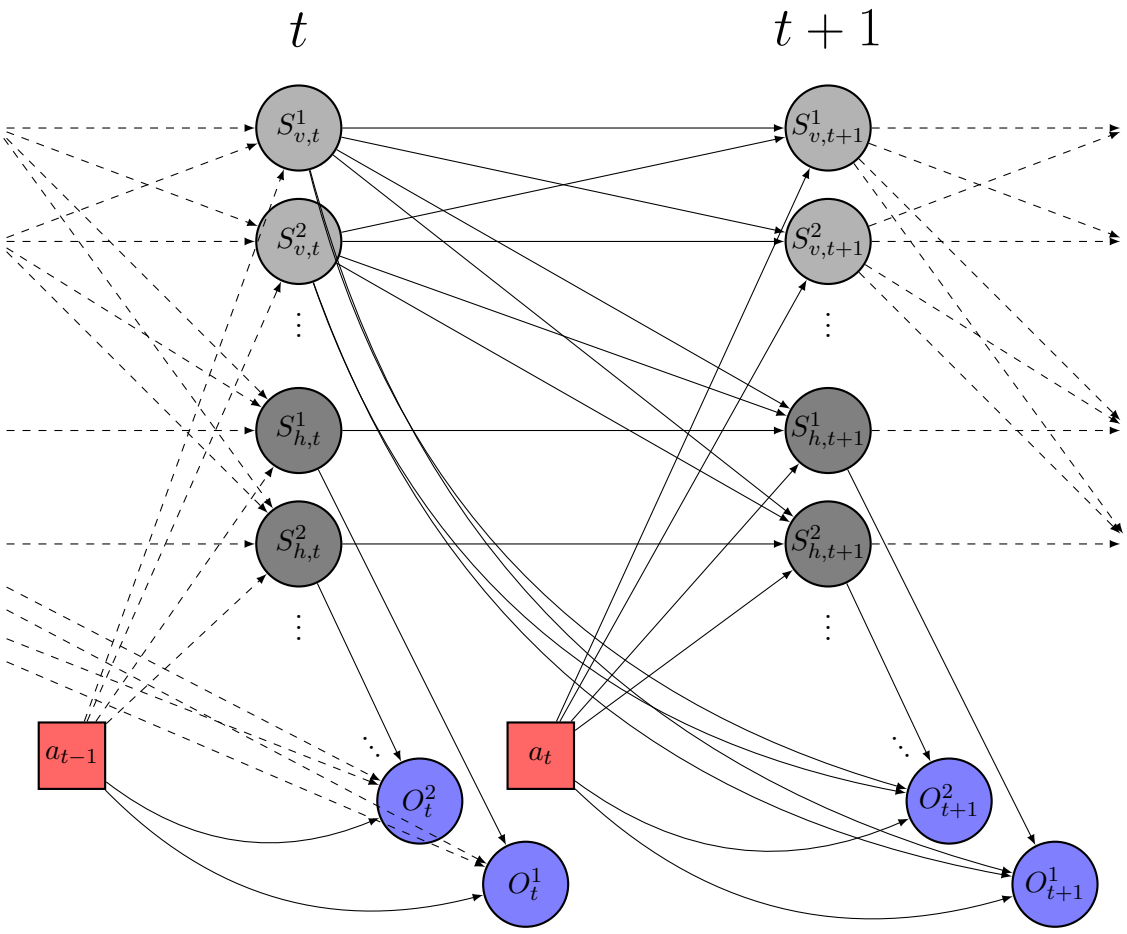


FIGURE III.4 – *DBN résumant les hypothèses d'indépendance d'un π-PDMOM menant à des variables de croyances marginales et un π-PDM avec une fonction de transition factorisée. Les parents d'une variable d'état visible sont les variables d'état visibles précédentes. Les parents d'une variable d'état cachée sont les variables d'état visibles précédentes et la variable d'état cachée précédente correspondante. Enfin, les parents d'une variable d'observation sont les variables d'état visibles précédentes, et la variables d'état cachée courante correspondante.*

$\mathcal{S}_v^m \times \Pi_{\mathcal{L}}^{\mathcal{S}_h^1} \times \cdots \times \Pi_{\mathcal{L}}^{\mathcal{S}_h^l}$ avec $\Pi_{\mathcal{L}}^{\mathcal{S}_h^j} \subsetneq \mathcal{L}^{\mathcal{S}_h^j}$. La taille de $\Pi_{\mathcal{L}}^{\mathcal{S}_h^j}$ est $\#\mathcal{L}^{\#\mathcal{S}_h^j} - (\#\mathcal{L}-1)^{\#\mathcal{S}_h^j}$ (*cf.* équation I.19). Si toutes les variables d'état sont binaires, $\#\Pi_{\mathcal{L}}^{\mathcal{S}_h^j} = 2\#\mathcal{L} - 1$ pour tout $1 \leqslant j \leqslant l$, ainsi $\#\mathcal{S}_v \times \Pi_{\mathcal{L}}^{\mathcal{S}_h} = 2^m(2\#\mathcal{L}-1)^l$: contrairement au cadre probabiliste, **les états cachés et les états visibles ont un impact similaire sur la complexité de résolution**, *i.e.* tous les deux simplement exponentiels en le nombre de variables d'état. Dans le cas général, en notant $\kappa = \max\{\max_{1\leqslant i\leqslant m} \#S_{v,i}, \max_{1\leqslant j\leqslant l} \#S_{h,j}\}$, il y a $\mathcal{O}(\kappa^m(\#\mathcal{L})^{(\kappa-1)l})$ états de croyances, ce qui est exponentiel en l'arité des variables d'état.

La **fonction de mise à jour de l'état de croyance marginal** est ν^j :

$$\beta_{h,t+1}^j = \nu^j(s_{v,t}, \beta_{h,t}^j, a_t, o_{t+1}^j),$$

Cette mise à jour peut se noter

$$\beta_{h,t+1}^j(s_{h,t+1}^j) \propto^\pi \pi\left(o_{t+1}^j, s_{h,t+1}^j \;\middle|\; s_{v,t}, \beta_{h,t}^j, a_t\right)$$

puisqu'elle normalise au sens possibiliste la distribution de possibilité jointe sur la variable d'état cachée et la variable d'observation qui ont le numéro j.

Ainsi, le degré de possibilité que la variable de croyance marginale $B_{h,t+1}^{\pi,j}$ soit égale à $\beta_{h,t+1}^j \in \Pi_{\mathcal{L}}^{\mathcal{S}_h^j}$ conditionnellement à $B_{h,t}^{\pi,j} = \beta_{h,t}^j$ et l'action $a_t \in \mathcal{A}$, peut se calculer :

$$\Pi\left(B_{h,t+1}^{\pi,j} = \beta_{h,t+1}^j \;\middle|\; S_{v,t} = s_{v,t}, B_{h,t}^{\pi,j} = \beta_{h,t}^j, a_t\right) = \max_{\substack{o^j \in \mathcal{O}^j \text{ s.t.} \\ \nu^j(s_{v,t},\beta_{h,t}^j,a_t,o^j)=\beta_{h,t+1}^j}} \pi\left(o^j \;\middle|\; s_{v,t}, \beta_{h,t}^j, a_t\right) \tag{III.2}$$

définissant la distribution de possibilité de transition des états de croyance marginaux $\pi\left(\beta_{h,t+1}^j \;\middle|\; s_{v,t}, \beta_{h,t}^j, a_t\right)$.

Enfin, le théorème 9 nous assure que les variables du π-PDM résultant d'un tel π-PDMOM sont indépendantes post-action conditionnellement à l'état courant du système : cela permet alors d'écrire la fonction de transition de ce π-PDM sous forme factorisée :

Théoreme 9 (*Expression factorisée de la distribution de possibilité de transition*)

Si les hypothèses d'indépendance d'un π-PDMOM sont décrites par le DBN de la figure III.4, alors $\forall \beta_{h,t} = (\beta_{h,t}^1, \ldots, \beta_{h,t}^l) \in \Pi_{\mathcal{L}}^{\mathcal{S}_h}, \beta_{h,t+1} = (\beta_{h,t+1}^1, \ldots, \beta_{h,t+1}^l) \in \Pi_{\mathcal{L}}^{\mathcal{S}_h}$, $\forall (s_{v,t}, s_{v,t+1}) \in (\mathcal{S}_v)^2$, $\forall a_t \in \mathcal{A}$,

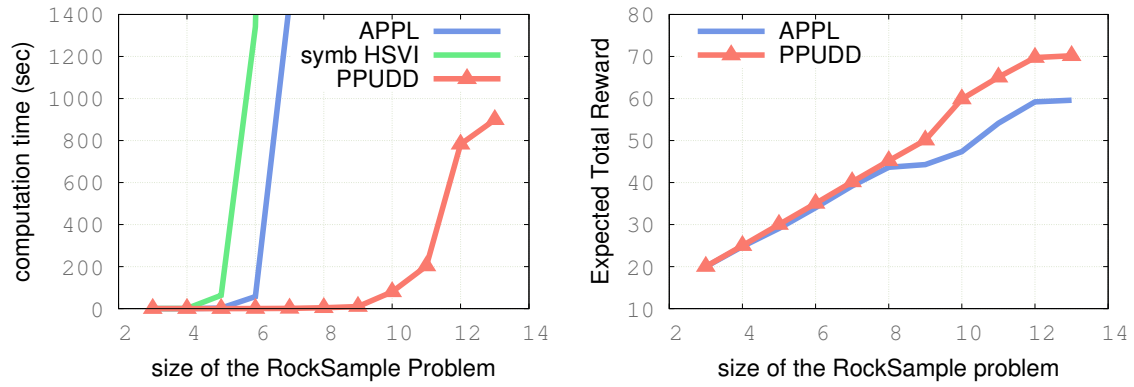
$$\pi\left(s_{v,t+1}, \beta_{h,t+1} \;\middle|\; s_{v,t}, \beta_{h,t}, a\right)$$

$$= \min\left\{\min_{i=1}^{m} \pi\left(s_{v,t+1}^i \;\middle|\; s_{v,t}, a_t\right), \min_{j=1}^{l} \pi\left(\beta_{h,t+1}^j \;\middle|\; s_{v,t}, \beta_{h,t}^j, a_t\right)\right\},$$

où la distribution de possibilité de transition des variables d'état visibles est donnée par l'équation (III.1) et celle des variables de croyance marginales par l'équation (III.2).

En utilisant ce résultat, une telle expression factorisée de la distribution de possibilité de transition permet le calcul d'une fonction valeur avec $n = m + l$ étapes, comme décrit dans la section précédente : le π-PDMPO est en effet un π-PDM factorisé puisque les variables $(S_v^1, \ldots, S_v^m, B_h^{\pi,1}, \ldots, B_h^{\pi,l})$, peuvent jouer le rôle des variables $X_1, \ldots, X_n$ dans l'algorithme 4.

Notons que la relation d'indépendance probabiliste satisfait les axiomes des semi-graphoides : ainsi, les résultats d'indépendance dus à la d-Séparation sont aussi vrais pour le cadre probabiliste. Si les hypothèses d'indépendance d'un PDMOM probabiliste [52, 2] sont décrites par le *DBN* de la figure III.4, alors une factorisation similaire peut être déduite.



(a) *Temps de calcul (secondes)* (b) *Récompense totale espérée*

FIGURE III.5 – *Comparaison entre PPUDD et les planificateurs probabilistes APPL et symb-HSVI sur le problème RockSample : l'axe des abscisses représente l'indice de l'instance du problème qui est croissant avec la complexité de instance du problème.*

Le PDM construit à partir d'un tel PDMOM probabiliste est donc un PDM factorisé, dont l'espace d'état est infini.

Les théorèmes précédents permettent d'écrire la fonction de transition d'un (π-)PDM résultant d'un (π-)PDMOM avec des distributions qui concernent moins de variables. La mise à jour de la fonction valeur durant la programmation dynamique est alors divisée en $n = m + l$ étapes dans le cas possibiliste, comme décrit par la boucle *for* de l'algorithme 4. Cette boucle permet de manipuler des *ADDs* qui ont moins de noeuds ce qui rend les calculs plus rapides en général. Ces résultats sont utilisés dans la section suivante afin de calculer de manière plus efficace les stratégies optimales d'un π-PDMOM structuré.

III.4 Résultats Expérimentaux

Les calculs sont beaucoup plus simples dans le cadre possibiliste, mais il faut évidemment en payer le prix : les modèles possibilistes peuvent être considérés comme des approximations de modèles probabilistes. Comme montré dans cette section, une approximation possibiliste en utilisant *PPUDD* peut mener à de meilleures stratégies, au sens probabiliste, que celles calculées par certains algorithmes probabilistes lorsque les dimensions du problème considéré rendent les calculs insurmontables.

III.4.1 Missions Robotiques

Nous avons comparé *PPUDD* sur le problème *Rocksample* (RS), décrit en section III.3.1, contre un récent planificateur probabiliste pour PDMOM, *APPL* [52], et un planificateur pour PDMPO utilisant des *ADDs*, *symbolic HSVI* [69]. Ces deux algorithmes peuvent être arrêtés à n'importe quel moment, et plus les calculs durent, plus la stratégie calculée est performante : ainsi nous avons décidé d'arrêter les calculs lorsque l'erreur d'approximation passe en-dessous de 1. La figure III.5a, où l'instance du problème augmente avec la complexité du problème, montre que *APPL* déborde en mémoire à l'instance numéro 8, et *symbolic HSVI* à l'instance numéro 7, tandis que *PPUDD* peut calculer une stratégie pour des instances beaucoup plus compliquées. Nous pouvons aussi fixer une durée de calcul aux algorithmes probabilistes utilisés : le temps de calcul d'*APPL* est alors fixé au temps de calcul de *PPUDD*, et les performances de leurs stratégies respectives sont comparées en terme d'espérances de la somme des récompenses (et en utilisant les récompenses définies par le modèle probabiliste). Étonnam-

ment, la figure III.5b montre que les récompenses récupérées sont en moyenne plus grandes avec *PPUDD* qu'avec *APPL*. La raison est que *APPL* est en fait un planificateur probabiliste qui cherche à raffiner une approximation durant le temps de calcul, ce qui montre qui notre approche consistant à résoudre exactement un modèle approximé peut mieux fonctionner que de résoudre de manière approchée un modèle exact. Enfin, nous pouvons noter que les probabilités du modèle d'observation, qui représentent les incertitudes concernant les réponses des capteurs, peut être difficile à connaître précisément en pratique, auquel cas les modèles possibilistes peuvent plus physiquement rigoureux.

Ces résultats nous ont permis de juger pertinent la participation de *PPUDD* à la compétition internationale de planification probabiliste 2014, même si le calcul de stratégie pour les modèles probabilistes n'est pas la vocation initiale de ce planificateur. La section suivante discute des résultats des différents planificateurs.

III.4.2 Compétition Internationale de Planification Probabiliste 2014

La session entièrement observable de la compétition internationale de planification probabiliste permet de comparer des planificateurs de PDM en garantissant les même ressources en terme de puissance de calcul et de temps de calcul à chacun des algorithmes participants. Les planificateurs compétiteurs doivent calculer des stratégies pour certains problèmes qui ne sont pas connus à l'avance. Étant donné un de ces problèmes, les planificateurs ont un temps limité pour envoyer des actions à un serveur de la compétition qui simule l'évolution de l'état du système : les états successifs sont générés par le serveur de la compétition en utilisant la fonction de transition probabiliste du PDM définissant le problème, et envoyés à un des serveurs de compétiteurs. Pour chaque état du système reçu, le planificateur du compétiteur doit envoyer une action, qu'il a calculé, en retour. Ces échanges de données se produisent sur plusieurs exécutions d'horizon fini et le score du planificateur pour les problèmes considérés est la moyenne, sur les exécutions, de la somme finie des récompenses obtenue sur la trajectoire de l'état du système.

Plus d'informations à propos de cette compétition sont disponibles sur le site officiel de la compétition `https://cs.uwaterloo.ca/~mgrzes/IPPC_2014/`. Les problèmes sont regroupés dans des *domaines*, qui sont des PDM dont un nombre fini de paramètres ne sont pas définis : le problème, ou PDM, utilisé en pratique durant la compétition est une *instance* d'un domaine, *i.e.* un domaine dont les paramètres ont été définis. Dans cette compétition, 8 domaines ont été proposés, appelé respectivement *Academic advising*, *Crossing traffic*, *Elevators*, *Skill teaching*, *Tamarisk*, *Traffic*, *Triangle tireworld* et *Wildfire*. La compétition consiste à évaluer les planificateurs sur 10 instances par domaine avec 30 exécutions de chaque instance et 18 minutes allouées par instance : elle dure donc 24 heures au total.

Quatre planificateurs ont été proposés pour cette compétition :

— *PROST* [38], basé sur l'algorithme *Upper Confidence bound applied to Trees* (UCT, [39]) ;
— *GOURMAND* [42, 41], basé sur l'algorithme *Labeled Real Time Dynamic Programming* (*LRTDP*, [8]) ;
— *symbolic LRTDP* [22] ;
— notre algorithme *PPUDD*.

Comme le score donné aux planificateurs ne dépend que des 40 premières étapes du processus, la version présentée de *PPUDD* est l'algorithme 4 avec la "condition du while" $\overline{U^*} \neq \overline{U^c}$ à la ligne 2 remplacée par la condition "numéro de l'itération $\leqslant 40$". Il augmente de manière incrémentale l'horizon de planification en maintenant un masque stocké sous forme d'un *BDD* (arbre de décision binaire, *i.e.* un *ADD* avec des feuilles dans $\{0, 1\}$) représentant les états atteignables à partir de l'état initial : le calcul de la fonction valeur courante est alors restreinte

aux états atteignables seulement. Bien que *PPUDD* soit un algorithme de calcul hors-ligne, nous avons aussi proposé *AnyTime PPUDD* (*ATPPUDD*), une version qui gère les temps de calcul comme décrit dans [42].

La bibliothèque utilisée pour faire les calculs entre *ADDs* s'appelle *CU Decision Diagram Package* (*CUDD*, `http://vlsi.colorado.edu/~fabio/CUDD/`), et les versions de *PPUDD* décrites sont disponibles à l'adresse `https://github.com/drougui/ppudd`.

Les figures qui suivent sont les résultats d'*IPPC* 2014 : les scores sont donnés en fonction de l'indice de l'instance, qui augmente généralement avec la complexité du problème associé.

La figure III.6 présente les scores obtenus par chacun des planificateurs pour chacune des 10 instances du domaine *Academic advising*, *i.e.* la moyenne sur les 30 exécutions, de la somme des récompenses rencontrées. Les performances de notre algorithme sont proches de celles des meilleurs algorithmes. Cependant, un bug non expliqué et indésirable s'est produit avec *ATPPUDD* lors de l'instance numéro 2, puisque seules trois exécutions ont été possibles avec ce planificateur. Pour les autres instances, *PPUDD* et *ATPPUDD* produisent des stratégies dont les performances sont semblables à celles de PROST et GOURMAND, et meilleures que *Symbolic LRTDP*. Ceci n'est plus vrai avec le problème *Crossing traffic*, dont les résultats sont décrits par la figure III.6. Ce problème modélise un robot qui doit atteindre un but qui est de l'autre côté d'une route à plusieurs voies que de nombreuses voitures empruntent. Ces voitures arrivent de manière aléatoire et vont vers la gauche. Comme le degré de possibilité attribué au fait qu'aucune voiture arrive a été fixé à 1 par notre traduction naïve de PDM en π-PDM, le critère optimiste mène à la décision de traverser la route même si une voiture (invisible au moment de la décision) peut arriver sur la droite (avec une probabilité < 0.5 mais assez grande pour devoir plutôt être prudent, ou pessimiste, et traverser la route dans une position avec une visibilité sur l'arrivée des voitures). Ceci explique la pauvre qualité des stratégies produites par *PPUDD* pour ce domaine. Notons cependant que, pour les 6 dernières intances (*i.e.* les problèmes les plus difficiles) notre approche mène à de meilleures stratégies que le planificateur probabiliste *Symbolic LRTDP*.

Le problème *Elevators* concerne des gens qui arrivent de manière aléatoire devant un ascenseur et qui doivent être transportés à l'étage qu'ils ont choisi d'un immeuble : comme l'information fréquentiste est perdue lors de l'utilisation de l'approche possibiliste, et semble très importante dans ce problème (les gens ne veulent pas attendre une fois arrivés devant l'ascenseur), cela explique pourquoi les scores de notre algorithme sont moins bons que ceux de PROST et GOURMAND. Cependant *PPUDD* et *ATPPUDD* sont meilleurs que Symbolic LRTDP, comme montré dans la figure III.7, et la stratégie qui consiste à ne rien faire ("noop") ou la stratégie qui choisit des actions de manière aléatoire ("random"). *PPUDD* et *ATPPUDD* ont des comportements plutôt bons avec le problème *Skill teaching* comme illustré par la même figure. De plus, *ATPPUDD* mène à de meilleurs résultats pour les trois dernières instances : puisque ces instances sont celles du domaine *Skill teaching* avec l'espace d'état le plus grand, la version anytime, qui gère les temps de calculs de manière plus recherchée, produit des stratégies avec de meilleures performances que *PPUDD*, qui résout classiquement le problème du π-PDM, associé, mais ne peut pas terminer les calculs et mène à des stratégies moins bonnes.

Par rapport aux autres planificateurs, les algorithmes possibilistes donnent de bon résultats avec le domaine *Tamarisk*, comme le montre la figure III.8. Cependant, certaines instances (par exemple les instances numéro 6, 8 et 10) ne sont même pas exécutées puisque l'instanciation des *ADDs* définissant le problème prennent trop de temps. *Symbolic LRTP* fait face au mêmes problèmes puisqu'il utilise aussi des *ADDs*. Le domaine *Traffic* est vraiment très dur à résoudre avec *PPUDD* et *ATPPUDD* (*cf.* figure III.8). En fait, les pires scores sont obtenus avec ce domaine, et même la stratégie aléatoire ou la stratégie "noop" sont meilleures. Il aurait été avantageux pour nous d'implémenter un garde-fou renvoyant des actions aléatoires lorsque la stratégie calculée est moins performante que la stratégie aléatoire. Comme mentionné au-dessus

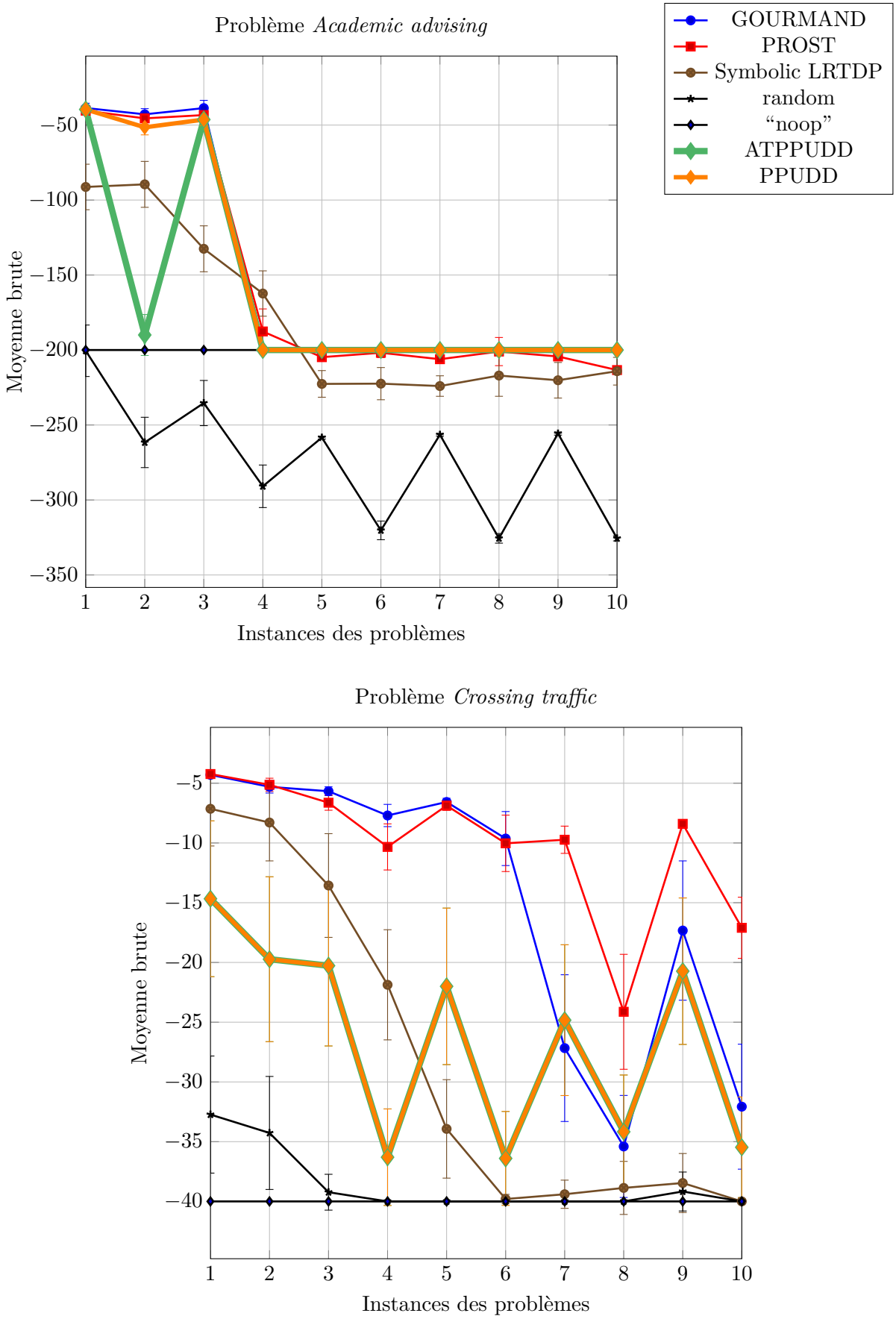


FIGURE III.6 – *Résultats de la compétition internationale de planification probabiliste – session entièrement observable*

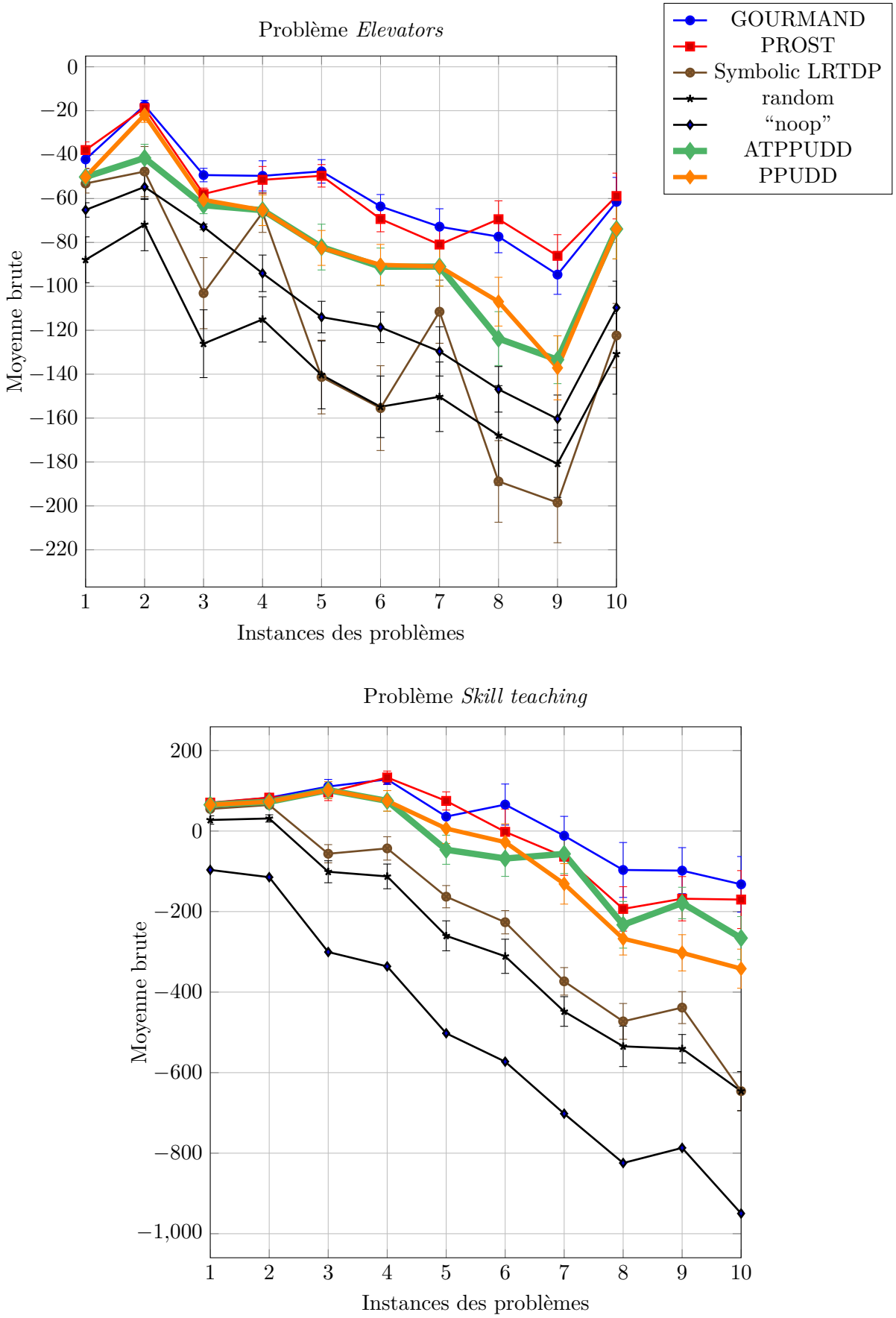


FIGURE III.7 – *Résultats de la compétition internationale de planification probabiliste – session entièrement observable*

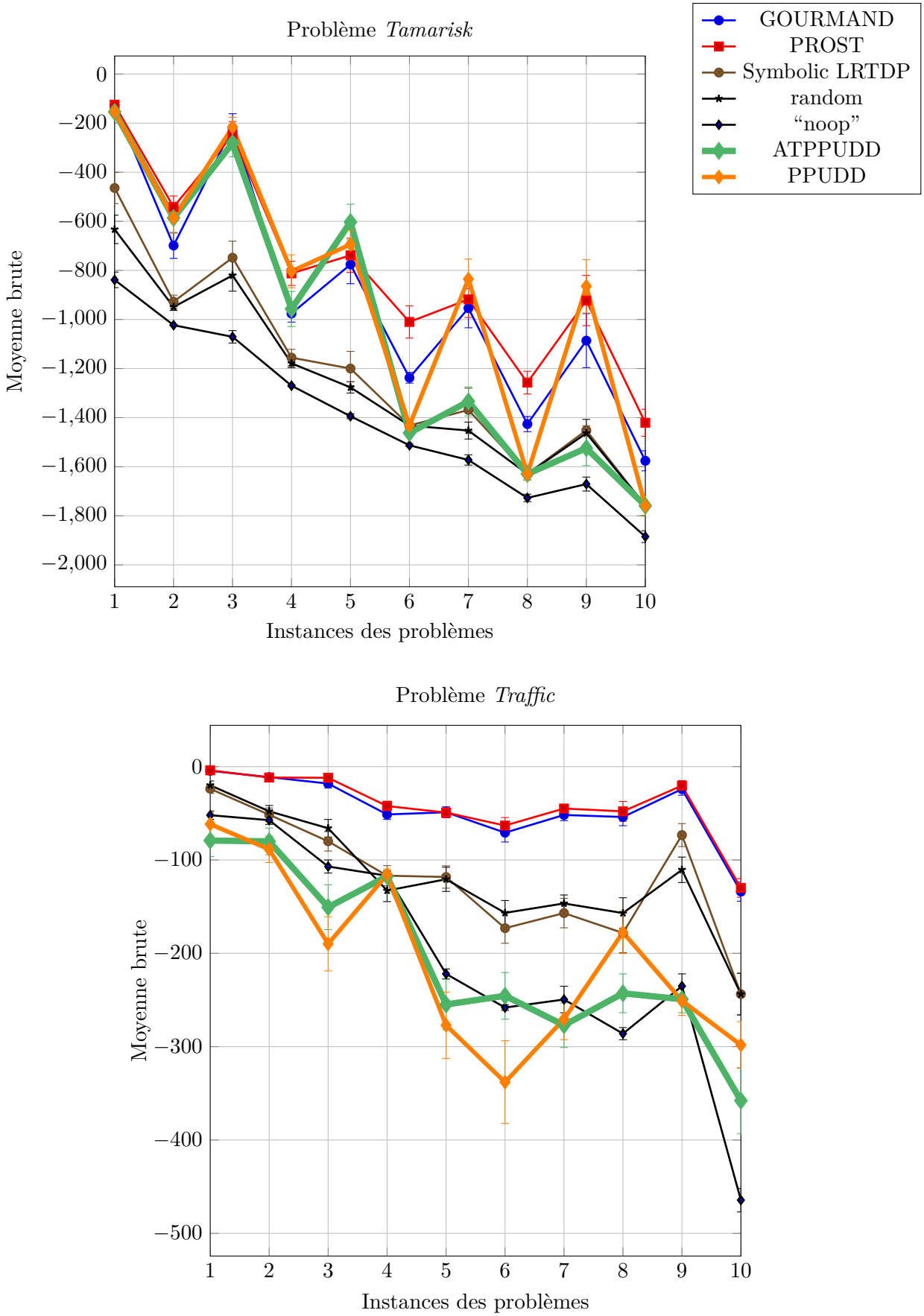


FIGURE III.8 – *Résultats de la compétition internationale de planification probabiliste – session entièrement observable*

avec le problème *Crossing traffic*, le critère optimiste peut mener à des actions dangereuses, comme cela se produit ici. De plus, puisque ce problème implique des informations fréquentistes (arrivées des voitures aléatoires), notre approche est sûrement inadaptée pour cet exemple. Enfin, le domaine *Traffic* est connu comme étant un des domaines les plus durs à résoudre, ainsi l'instanciation des *ADDs* en mémoire prennent trop de temps, tout comme les calculs, qui ne sont alors pas assez avancés pour produire des résultats satisfaisants.

Finalement, les deux derniers domaines, dont les résultats sont décrits dans la figure III.9, sont appelés *Triangle Tireworld* et *Wildfire*. Tout d'abord, *ATPPUDD* fait face à un bug inexpliqué pour chaque instance du domaine *Triangle Tireworld* : aucune exécution n'est réalisée à partir de l'instance numéro 5, et deux exécutions ont été réalisées pour les autres instances (ce qui explique le faible score pour chaque instance). Comme déjà mentionné pour le domaine *Tamarisk*, l'instanciation des *ADDs* prend trop de temps pour les dernières instances, et aucune exécution n'est réalisée pour les 4 dernières instances avec *PPUDD* : *Symbolic LRTDP* rencontre les même problèmes. Le dernier domaine, appelé *Wildfire*, constitue un problème très fréquentiste : il implique des départs de feux. C'est pourquoi *PPUDD* et *ATPPUDD* ne sont pas vraiment efficaces, mais pas trop distant non plus des résultats de *Symbolic LRTDP*.

III.5 Conclusion

Nous avons présenté *PPUDD*, le premier algorithme, à notre connaissance, qui résout les PDM(OM) possibilistes qualitatifs avec des calculs symboliques. Nous pensons que les modèles possibilistes constituent un bon compromis entre les modèles non-déterministes, où l'incertitude n'est pas quantifiée du tout, menant à un modèle très approximatif, et les modèles probabilistes, où l'incertitude est complètement spécifiée, et parfois de manière arbitraire en pratique. La résolution de problèmes de planification en utilisant le cadre du non-déterminisme est appelé *planification contingente/conformante*, étudiée par exemple dans [1, 9].

Nos résultats expérimentaux montrent que l'utilisation d'un algorithme exact (*PPUDD*) pour résoudre un modèle approximé (π-PDMOM) peut mener à des calculs plus rapides que de raisonner sur des modèles exacts et complexes, tout en générant des stratégies qui peuvent être de meilleure qualité que celles retournées par des algorithmes procédant à des approximations (*APPL*) sur des modèles exacts (PDMOM).

Enfin, ce chapitre présente les résultats de notre approche possibiliste durant *IPPC* 2014 : un des problèmes de cette approche est la traduction du modèle probabiliste en modèle possibiliste : la traduction automatique naïve utilisée est peut être à l'origine des mauvaises performances de l'approche pour certain domaines à la dynamique, ou la fonction de récompense complexe. Un autre problème semble être l'utilisation des *ADDs* : l'instanciation de ces objets avant même le début des calculs prend un temps trop important, où bien ne passe même pas en mémoire. Cette difficulté est partagée avec le planificateur *Symbolic LRTDP*. Des problèmes de modélisation ont aussi été mis en évidence : certains problèmes nécessitent une approche optimiste, et d'autres une approche pessimiste.

Cependant, bien que PROST et GOURMAND ont des performances deux fois supérieures à *PPUDD* et *ATPPUDD*, ces derniers proposent des stratégies dont les performances sont deux fois supérieures à son homologue probabiliste *Symbolic LRTDP*. Notons finalement que *PPUDD* est disponible dans le dépôt `https://github.com/drougui/ppudd`.

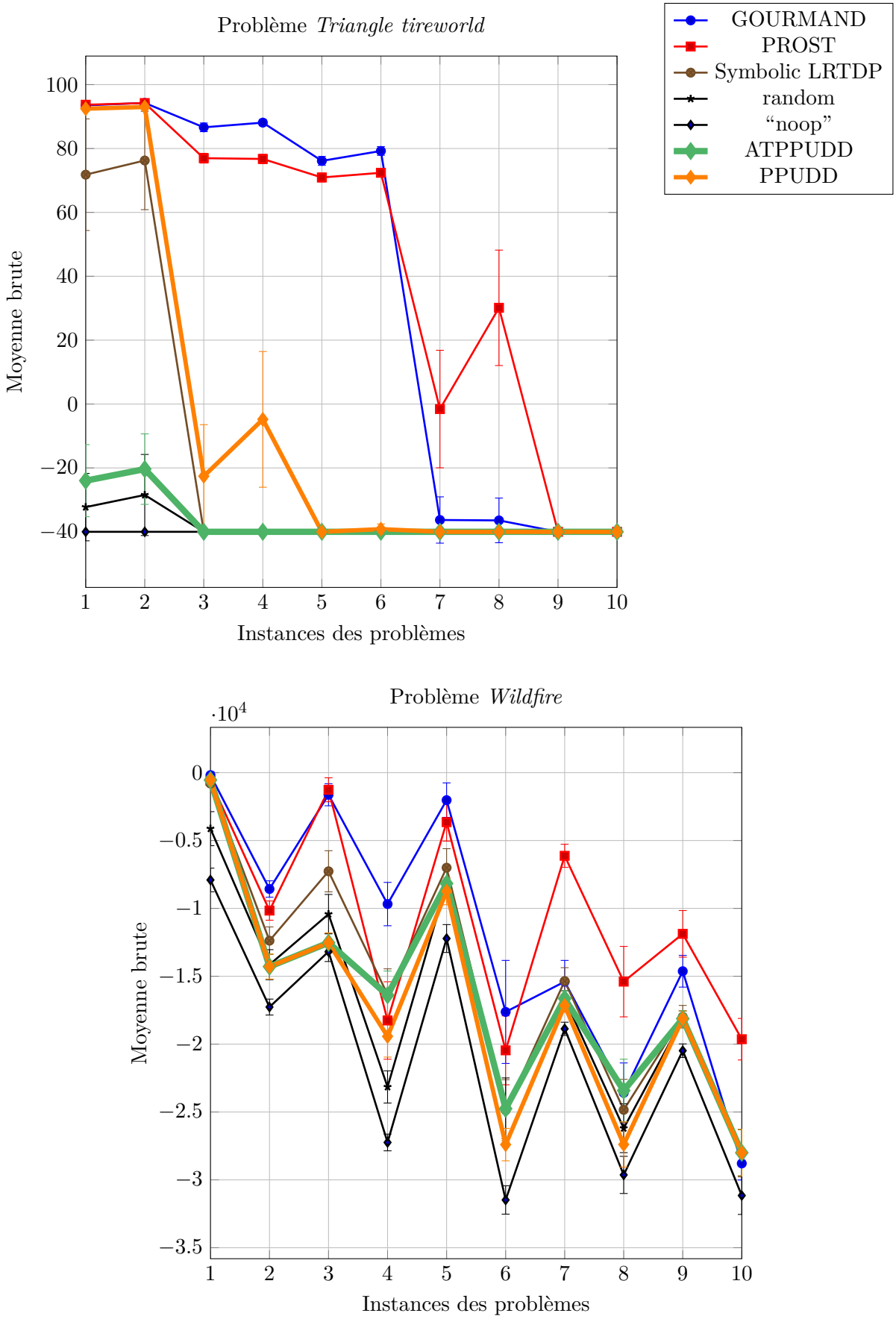


FIGURE III.9 – *Résultats de la compétition internationale de planification probabiliste – session entièrement observable*

Conclusion

Les contributions de cette thèse sont principalement centrées autour des travaux préliminaire de Régis Sabbadin [62]. Ce dernier propose un homologue possibiliste qualitatif aux PDMPO modélisant l'incertitude avec des distributions de possibilité qualitatives. Ainsi, dans notre travail, des modèles possibilistes qualitatifs pour la planification sous incertitude ont été développés et étudiés. Cette étude a été motivée par différents problèmes posés par les PDMPO détaillés en introduction.

La motivation majeure de cette étude est la réduction de la complexité des calculs offerte par les π-PDMPO : son espace d'états de croyance est fini, tandis que l'espace des états de croyance probabilistes est infini. De plus, comme ses états de croyance sont des distributions de possibilité sur l'espace des états, l'ignorance totale peut être définie par une distribution de possibilité égale à 1 sur tous les états. Si un état donné du système est parfaitement connu comme étant l'état courant, l'état de croyance associé attribue 1 à cet état du système et 0 à tous les autres puisqu'ils sont impossibles : c'est une représentation plus appropriée que la probabilité uniforme. L'imprécision des distributions de probabilités est aussi naturellement gérée par le formalisme possibiliste qualitatif.

Plus généralement, les modèles possibilistes qualitatifs ont besoin de moins d'information à propos du système que le modèle probabiliste : les plausibilités des événements sont "seulement" classifiées dans l'échelle possibiliste $\mathcal{L}$ plutôt que quantifiées. Cette thèse propose finalement des contributions théoriques et pratiques à propos de ces processus possibilistes qualitatifs : d'une part, les contributions théoriques sont par exemple l'observabilité mixte, la gestion des horizons indéfinis, ou les résultats d'indépendance. D'autre part, la principale contribution pratique est la démonstration que ces modèles possibilistes qualitatifs ont un intérêt dans la simplification des calculs ou la modélisation via des résultats expérimentaux (par exemple IPPC 2014) ou l'étude du comportement de la croyance possibiliste.

Ces contributions sont amenées à travers des applications robotiques afin de faire le lien avec les problématiques de départ : des missions de reconnaissance de cible sont étudiées (par exemple le problème *Rocksample*, ou même la mission décrite par la figure II.2) ; IPPC 2014 contient aussi des problèmes comparables à des systèmes robotiques (par exemple le problème *Elevator*, ou le problème *Tamarisk*, aussi utilisé lors de la compétition d'apprentissage par renforcement 2014).

Perspectives

La première perspective vient d'une observation : les problèmes de mémoire atteints avec le domaine *Triangle tireworld* d'IPPC 2014 pourrait être contourné à l'aide d'une discrétisation plus importante : les version de *PPUDD* pour IPPC 2014 utilisaient une précision de 10^{-3} sur le problème probabiliste initial, menant à une importante échelle possibiliste $\mathcal{L}$. Il serait instructif d'avoir une idée de l'impact de ce prétraitement sur les performances de *PPUDD* : plus généralement, un travail plus important sur la traduction d'un PDM probabiliste en approximation possibiliste pourrait améliorer de manière significative l'approche proposée pour IPPC 2014.

En terme de méthode de calculs, nous nous sommes concentrés sur des résolutions symboliques des π-PDM (*PPUDD*). Cependant, des méthodes de résolutions alternatives pourraient être étudiées : par exemple, des méthodes heuristiques, qui sont efficaces pour les problèmes probabilistes [73], pourraient être adaptées au contexte possibiliste. En ce qui concerne l'apprentissage par renforcement dans le cadre des possibilités qualitatives, nous pouvons citer le travail suivant [65].

Enfin, les avancées de la théorie des possibilités peuvent permettre de raffiner les PDM possibilistes qualitatifs afin d'améliorer la modélisation lorsque c'est nécessaire. Nous pouvons par exemple mentionner l'opérateur *leximin* [26] : il permet d'éviter l'effet de noyade des opérateurs min et max. Il peut être utilisé pour l'aggrégation de préférences ou pour calculer un degré de possibilité plus fin d'une trajectoire. Notons que cela peut être une amélioration utile, mais qu'elle complexifie inévitablement le problème. Des critères plus discriminants sont aussi étudiés dans la littérature [76, 35].

Bibliographie

[1] Alexandre Albore, Héctor Palacios, and Hector Geffner. A translation-based approach to contingent planning. In Craig Boutilier, editor, *IJCAI*, pages 1623–1628, 2009. (Cité page 50.)

[2] M. Araya-Lòpez, V. Thomas, O. Buffet, and F. Charpillet. A closer look at MOMDPs. In *Proceedings of the Twenty-Second IEEE International Conference on Tools with Artificial Intelligence (ICTAI-10)*, 2010. (Cité pages 13, 28, 31 et 43.)

[3] R. I. Bahar, E. A. Frohm, C. M. Gaona, G. D. Hachtel, E. Macii, A. Pardo, and F. Somenzi. Algebric decision diagrams and their applications. *Form. Methods Syst. Des.*, 10(2-3) :171–206, April 1997. (Cité pages 35 et 38.)

[4] Richard Bellman. The theory of dynamic programming. *Bull. Amer. Math. Soc.*, 60(6) :503–515, 11 1954. (Cité page 9.)

[5] Richard Bellman. A Markovian Decision Process. *Indiana Univ. Math. J.*, 6 :679–684, 1957. (Cité page 3.)

[6] Nahla Ben Amor. *Qualitative possibilistic graphical models : from independence to propagation algorithms*. Thèse de doctorat, ISG - Université de Tunis, Tunis, juin 2002. (Cité pages 13 et 19.)

[7] Blai Bonet. New grid-based algorithms for partially observable Markov decision processes : Theory and practice. (Cité page 7.)

[8] Blai Bonet. Labeled RTDP : improving the convergence of real-time dynamic programming. In *In Proc. ICAPS-03*, pages 12–21. AAAI Press, 2003. (Cité page 45.)

[9] Blai Bonet and Hector Geffner. Flexible and scalable partially observable planning with linear translations. In *AAAI*, 2014. (Cité page 50.)

[10] Christian Borgelt, Jörg Gebhardt, and Rudolf Kruse. Graphical models. In *In Proceedings of International School for the Synthesis of Expert Knowledge (ISSEK'98*, pages 51–68. Wiley, 2002. (Cité page 13.)

[11] Craig Boutilier, Richard Dearden, and Moisés Goldszmidt. Stochastic dynamic programming with factored representations. *Artif. Intell.*, 121(1-2) :49–107, 2000. (Cité page 35.)

[12] Xavier Boyen and Daphne Koller. Exploiting the architecture of dynamic systems. In *AAAI/IAAI*, pages 313–320, 1999. (Cité page 37.)

[13] Ronen I. Brafman. A heuristic variable grid solution method for POMDPs. In *In AAAI*, pages 727–733, 1997. (Cité page 7.)

[14] Anthony Cassandra, Michael L. Littman, and Nevin L. Zhang. Incremental pruning : A simple, fast, exact method for partially observable Markov decision processes. In *In Proceedings of the Thirteenth Conference on Uncertainty in Artificial Intelligence*, pages 54–61. Morgan Kaufmann Publishers, 1997. (Cité page 7.)

[15] Caroline Ponzoni Carvalho Chanel, Florent Teichteil-Königsbuch, and Charles Lesire. POMDP-based online target detection and recognition for autonomous UAVs. In *ECAI*

2012 - 20th European Conference on Artificial Intelligence. Including Prestigious Applications of Artificial Intelligence (PAIS-2012) System Demonstrations Track, Montpellier, France, August 27-31 , 2012, pages 955–960, 2012. (Cité page 6.)

[16] Caroline Ponzoni Carvalho Chanel, Florent Teichteil-Königsbuch, and Charles Lesire. Multi-target detection and recognition by UAVs using online POMDPs. In *Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence, July 14-18, 2013, Bellevue, Washington, USA.*, 2013. (Cité page 6.)

[17] Gustave Choquet. Theory of capacities. *Annales de l'institut Fourier*, 5 :131–295, 1954. (Cité page 17.)

[18] Ronan Collobert, Koray Kavukcuoglu, and Clément Farabet. Torch7 : A Matlab-like environment for machine learning. In *BigLearn, NIPS Workshop*, 2011. (Cité page 9.)

[19] Ines Couso and Sébastien Destercke. Didier's groundhog day. 19(2) :10–15, December 2012. (Cité page 15.)

[20] B. De Finetti. *Theory of probability : a critical introductory treatment.* Wiley series in probability and mathematical statistics. Probability and mathematical statistics. Wiley, 1974. (Cité pages 10 et 15.)

[21] Thomas Dean and Keiji Kanazawa. A model for reasoning about persistence and causation. *Comput. Intell.*, 5(3) :142–150, December 1989. (Cité page 37.)

[22] KarinaValdivia Delgado, Cheng Fang, Scott Sanner, and Leliane Nunes de Barros. Symbolic bounded real-time dynamic programming. In Antônio Carlos da Rocha Costa, Rosa Maria Vicari, and Flavio Tonidandel, editors, *Advances in Artificial Intelligencev - SBIA 2010*, volume 6404 of *Lecture Notes in Computer Science*, pages 193–202. Springer Berlin Heidelberg, 2010. (Cité page 45.)

[23] Nicolas Drougard, Didier Dubois, Jean-Loup Farges, and Florent Teichteil-Königsbuch. Planning in partially observable domains with fuzzy epistemic states and probabilistic dynamics. In *Scalable Uncertainty Management - 9th International Conference, SUM 2015, Québec City, QC, Canada, September 16-18, 2015. Proceedings*, pages 220–233, 2015. (Cité page 14.)

[24] Nicolas Drougard, Florent Teichteil-Konigsbuch, Jean-Loup Farges, and Didier Dubois. Qualitative possibilistic mixed-observable MDPs. In *Proceedings of the Twenty-Ninth Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-13)*, pages 192–201, Corvallis, Oregon, 2013. AUAI Press. (Cité page 13.)

[25] Nicolas Drougard, Florent Teichteil-Königsbuch, Jean-Loup Farges, and Didier Dubois. Structured possibilistic planning using decision diagrams. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, July 27 -31, 2014, Québec City, Québec, Canada.*, pages 2257–2263, 2014. (Cité page 14.)

[26] Didier Dubois and HÃ©lÃ¨ne Fargier. Lexicographic refinements of sugeno integrals. In Khaled Mellouli, editor, *Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, volume 4724 of *Lecture Notes in Computer Science*, pages 611–622. Springer Berlin Heidelberg, 2007. (Cité page 54.)

[27] Didier Dubois and Henri Prade. *Possibility Theory : An Approach to Computerized Processing of Uncertainty (traduction revue et augmentée de "Théorie des Possibilités")*. Plenum Press, New York, 1988. (Cité page 15.)

[28] Didier Dubois and Henri Prade. Possibility Theory as a basis for qualitative decision theory. In *IJCAI*, pages 1924–1930. Morgan Kaufmann, 1995. (Cité pages 11 et 18.)

[29] Didier Dubois, Henri Prade, and Régis Sabbadin. Decision-theoretic foundations of qualitative possibility theory. *European Journal of Operational Research*, 128(3) :459–478, 2001. (Cité pages 12 et 18.)

[30] Didier Dubois, Henri Prade, and Sandra Sandri. On possibility/probability transformations. In *Proceedings of Fourth IFSA Conference*, pages 103–112. Kluwer Academic Publ, 1993. (Cité page 31.)

[31] Didier Dubois, Henri Prade, and Philippe Smets. Representing partial ignorance. *IEEE Trans. on Systems, Man and Cybernetics*, 26 :361–377, 1996. (Cité page 10.)

[32] Hélène Fargier, Nahla Ben Amor, and Wided Guezguez. On the complexity of decision making in possibilistic decision trees. *CoRR*, abs/1202.3718, 2012. (Cité page 12.)

[33] Laurent Garcia and Régis Sabbadin. Complexity results and algorithms for possibilistic influence diagrams. *Artificial Intelligence*, 172(8-9) :1018 – 1044, 2008. (Cité page 12.)

[34] Hector Geffner and Blai Bonet. Solving large POMDPs using real time dynamic programming. In *In Proc. AAAI Fall Symp. on POMDPs*, 1998. (Cité page 7.)

[35] Phan Hong Giang and Prakash P. Shenoy. A comparison of axiomatic approaches to qualitative decision making using possibility theory. In Jack S. Breese and Daphne Koller, editors, *UAI*, pages 162–170. Morgan Kaufmann, 2001. (Cité page 54.)

[36] Jesse Hoey, Robert St-Aubin, Alan Hu, and Craig Boutilier. SPUDD : Stochastic planning using decision diagrams. In *In Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, pages 279–288. Morgan Kaufmann, 1999. (Cité pages 14, 35, 37 et 38.)

[37] Hideaki Itoh and Kiyohiko Nakamura. Partially observable Markov decision processes with imprecise parameters. *Artificial Intelligence*, 171(8-9) :453 – 490, 2007. (Cité page 9.)

[38] Thomas Keller and Patrick Eyerich. PROST : probabilistic planning based on UCT. In *Proceedings of the Twenty-Second International Conference on Automated Planning and Scheduling, ICAPS 2012, Atibaia, São Paulo, Brazil, June 25-19, 2012*, 2012. (Cité pages 14 et 45.)

[39] Levente Kocsis and Csaba Szepesvári. Bandit based Monte-Carlo planning. In *Proceedings of the 17th European Conference on Machine Learning*, ECML'06, pages 282–293, Berlin, Heidelberg, 2006. Springer-Verlag. (Cité page 45.)

[40] Daphne Koller and Nir Friedman. *Probabilistic Graphical Models : Principles and Techniques - Adaptive Computation and Machine Learning*. The MIT Press, 2009. (Cité page 13.)

[41] Andrey Kolobov, Peng Dai, Mausam Daniel, and S. Weld. Reverse iterative deepening for finite-horizon mdps with large branching factors. In *In ICAPS'12*, 2012. (Cité page 45.)

[42] Andrey Kolobov, Mausam, and Daniel S. Weld. LRTDP versus UCT for online probabilistic planning. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence, July 22-26, 2012, Toronto, Ontario, Canada.*, 2012. (Cité pages 14, 45 et 46.)

[43] Hanna Kurniawati, David Hsu, and Wee Sun Lee. SARSOP : Efficient point-based POMDP planning by approximating optimally reachable belief spaces. In *Proceedings of Robotics : Science and Systems IV*, Zurich, Switzerland, June 2008. (Cité pages 7 et 36.)

[44] Steven M. LaValle. *Planning Algorithms*. Cambridge University Press, New York, NY, USA, 2006. (Cité page 29.)

[45] Y. Lecun, F. J. Huang, and L. Bottou. Learning methods for generic object recognition with invariance to pose and lighting. volume 2, 2004. (Cité page 8.)

[46] Yann Lecun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, pages 2278–2324, 1998. (Cité page 8.)

[47] Omid Madani, Steve Hanks, and Anne Condon. On the undecidability of probabilistic planning and infinite-horizon partially observable Markov decision problems. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence and the Eleventh Innovative Applications of Artificial Intelligence Conference Innovative Applications of Artificial Intelligence*, AAAI '99/IAAI '99, pages 541–548, Menlo Park, CA, USA, 1999. American Association for Artificial Intelligence. (Cité pages 6 et 12.)

[48] Bhaskara Marthi. Robust navigation execution by planning in belief space. In *Robotics : Science and Systems VIII, University of Sydney, Sydney, NSW, Australia, July 9-13, 2012*, 2012. (Cité page 6.)

[49] Martin Mundhenk. The complexity of planning with partially-observable Markov decision processes. Technical report, Hanover, NH, USA, 2000. (Cité page 7.)

[50] Yaodong Ni and Zhi-Qiang Liu. Policy iteration for bounded-parameter POMDPs. *Soft Computing*, pages 1–12, 2012. (Cité page 9.)

[51] Arnab Nilim and Laurent El Ghaoui. Robust control of Markov decision processes with uncertain transition matrices. *Oper. Res.*, 53(5) :780–798, September-October 2005. (Cité page 9.)

[52] Sylvie C. W. Ong, Shao Wei Png, David Hsu, and Wee Sun Lee. Planning under uncertainty for robotic tasks with mixed observability. *Int. J. Rob. Res.*, 29(8) :1053–1068, July 2010. (Cité pages 6, 13, 28, 31, 36, 43 et 44.)

[53] Takayuki Osogami. Robust partially observable Markov decision process. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, pages 106–115, 2015. (Cité page 9.)

[54] Christos Papadimitriou and John N. Tsitsiklis. The complexity of Markov decision processes. *Math. Oper. Res.*, 12(3) :441–450, August 1987. (Cité pages 6 et 12.)

[55] Judea Pearl. *Probabilistic reasoning in intelligent systems : networks of plausible inference*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1988. (Cité page 41.)

[56] Patrice Perny, Olivier Spanjaard, and Paul Weng. Algebraic Markov Decision Processes. In *19th International Joint Conference on Artificial Intelligence*, pages 1372–1377, 2005. (Cité page 21.)

[57] Joelle Pineau, Geoffrey Gordon, and Sebastian Thrun. Point-based value iteration : An anytime algorithm for POMDPs. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1025 – 1032, August 2003. (Cité page 7.)

[58] Joelle Pineau and Geoffrey J. Gordon. POMDP planning for robust robot control. In Sebastian Thrun, Rodney A. Brooks, and Hugh F. Durrant-Whyte, editors, *ISRR*, volume 28 of *Springer Tracts in Advanced Robotics*, pages 69–82. Springer, 2005. (Cité page 6.)

[59] Cédric Pralet, Thomas Schiex, and Gérard Verfaillie. *Sequential Decision-Making Problems - Representation and Solution*. Wiley, 2009. (Cité page 34.)

[60] Julia Radoszycki, Nathalie Peyrard, and Régis Sabbadin. Finding good stochastic factored policies for factored markov decision processes. In *ECAI 2014 - 21st European Conference on Artificial Intelligence, 18-22 August 2014, Prague, Czech Republic - Including Prestigious Applications of Intelligent Systems (PAIS 2014)*, pages 1083–1084, 2014. (Cité page 35.)

[61] Régis Sabbadin. *Une Approche Ordinale de la Decision dans l'Incertain : Axiomatisation, Representation Logique et Application à la Décision Séquentielle*. Thèse de doctorat, Université Paul Sabatier, Toulouse, France, décembre 1998. (Cité page 19.)

[62] Régis Sabbadin. A possibilistic model for qualitative sequential decision problems under uncertainty in partially observable environments. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, UAI'99, pages 567–574, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc. (Cité pages 11, 12, 13, 19, 21, 22, 29, 31 et 53.)

[63] Régis Sabbadin. Empirical comparison of probabilistic and possibilistic Markov decision processes algorithms. In Werner Horn, editor, *ECAI*, pages 586–590. IOS Press, 2000. (Cité pages 13 et 19.)

[64] Régis Sabbadin. Possibilistic Markov decision processes. *Engineering Applications of Artificial Intelligence*, 14(3) :287 – 300, 2001. Soft Computing for Planning and Scheduling. (Cité pages 13 et 19.)

[65] Régis Sabbadin. Towards possibilistic reinforcement learning algorithms. In *Proceedings of the 10th IEEE International Conference on Fuzzy Systems, Melbourne, Australia, December 2-5, 2001*, pages 404–407, 2001. (Cité page 54.)

[66] Régis Sabbadin, Hélène Fargier, and Jérôme Lang. Towards qualitative approaches to multi-stage decision making. *Int. J. Approx. Reasoning*, 19(3-4) :441–471, 1998. (Cité page 18.)

[67] Guy Shani, Pascal Poupart, Ronen I. Brafman, and Solomon Eyal Shimony. Efficient ADD operations for point-based algorithms. In *ICAPS*, pages 330–337, 2008. (Cité page 37.)

[68] David Silver and Joel Veness. Monte-carlo planning in large POMDPs. In J.D. Lafferty, C.K.I. Williams, J. Shawe-Taylor, R.S. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23*, pages 2164–2172. Curran Associates, Inc., 2010. (Cité page 7.)

[69] Hyeong Seop Sim, Kee-Eung Kim, Jin Hyung Kim, Du-Seong Chang, and Myoung-Wan Koo. Symbolic heuristic search value iteration for factored POMDPs. In *Proceedings of the 23rd National Conference on Artificial Intelligence - Volume 2*, AAAI'08, pages 1088–1093. AAAI Press, 2008. (Cité pages 36 et 44.)

[70] Richard D. Smallwood and Edward J. Sondik. *The Optimal Control of Partially Observable Markov Processes Over a Finite Horizon*, volume 21. INFORMS, 1973. (Cité page 4.)

[71] Trey Smith and Reid Simmons. Heuristic search value iteration for POMDPs. In *Proceedings of the 20th conference on Uncertainty in artificial intelligence*, UAI '04, pages 520–527, Arlington, Virginia, United States, 2004. AUAI Press. (Cité pages 7, 36 et 40.)

[72] M. Sugeno. *Theory of fuzzy integrals and its applications*. PhD thesis, Tokyo Institute of Technology, 1974. (Cité page 17.)

[73] Florent Teichteil-Königsbuch, Vincent Vidal, and Guillaume Infantes. Extending classical planning heuristics to probabilistic planning with dead-ends. In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2011, San Francisco, California, USA, August 7-11, 2011*, 2011. (Cité page 54.)

[74] Judea P. Thomas Verma. Influence Diagrams and d-Separation. Technical report, July 1988. (Cité pages 14 et 41.)

[75] Tom S. Verma and Judea Pearl. Causal networks : Semantics and expressiveness. *CoRR*, abs/1304.2379, 2013. (Cité page 41.)

[76] Paul Weng. Qualitative Decision-Making Under Possibilistic Uncertainty : Toward More Discriminating Criteria. In *21st International Conference on Uncertainty in Artificial Intelligence*, volume 21, pages 615–622, 2005. INT LIP6 DECISION. (Cité page 54.)

[77] Paul Weng. Conditions générales pour l'admissibilité de la programmation dynamique dans la décision séquentielle possibiliste. *Revue d'Intelligence Artificielle*, 21(1) :129–143, 2007. NAT LIP6 DECISION. (Cité pages 13 et 34.)

[78] Stefan Witwicki, Francisco S. Melo, Jesús Capitán, and Matthijs T. J. Spaan. A flexible approach to modeling unpredictable events in MDPs. In *Proc. of Int. Conf. on Automated Planning and Scheduling*, pages 260–268, 2013. (Cité page 41.)

[79] L.A. Zadeh. Fuzzy sets. *Information and Control*, 8(3) :338 – 353, 1965. (Cité page 15.)

Title: Exploiting Imprecise Information Sources for Sequential Decision Making under Uncertainty

Abstract: Partially Observable Markov Decision Processes (POMDPs) define a useful formalism to express probabilistic sequential decision problems under uncertainty. When this model is used for a robotic mission, the *system* is defined as the features of the robot and its environment, needed to express the mission. The system state is not directly seen by the agent (the robot). Solving a POMDP consists thus in computing a strategy which, on average, achieves the mission best *i.e.* a function mapping the information known by the agent to an action. Some practical issues of the POMDP model are first highlighted in the robotic context: it concerns the modeling of the agent ignorance, the imprecision of the observation model and the complexity of solving real world problems. A counterpart of the POMDP model, called π-POMDP, simplifies uncertainty representation with a qualitative evaluation of event plausibilities. It comes from Qualitative Possibility Theory which provides the means to model imprecision and ignorance. After a formal presentation of the POMDP and π-POMDP models, an update of the possibilistic model is proposed. Next, the study of factored π-POMDPs allows to set up an algorithm named PPUDD which uses Algebraic Decision Diagrams to solve large structured planning problems. Strategies computed by PPUDD, which have been tested in the context of the competition IPPC 2014, can be more efficient than those produced by probabilistic solvers when the model is imprecise or for high dimensional problems. We show next that the π-Hidden Markov Processes (π-HMP), *i.e.* π-POMDPs without action, produces useful diagnosis in the context of Human-Machine interactions. Finally, a hybrid POMDP benefiting from the possibilistic and the probabilistic approach is built: the qualitative framework is only used to maintain the agent's knowledge. This leads to a strategy which is pessimistic facing the lack of knowledge, and computable with a solver of fully observable Markov Decision Processes (MDPs). This thesis proposes some ways of using Qualitative Possibility Theory to improve computation time and uncertainty modeling in practice.

Keywords: POMDP, Planning under Uncertainty, Possibility Theory, Autonomous Robotics, Imprecise Knowledge

Titre: Tirer Profit de Sources d'Information Imprécises pour la Décision Séquentielle dans l'Incertain

Résumé: Les Processus Décisionnels de Markov Partiellement Observables (PDMPOs) permettent de modéliser facilement les problèmes probabilistes de décision séquentielle dans l'incertain. Lorsqu'il s'agit d'une mission robotique, les caractéristiques du robot et de son environnement nécessaires à la définition de la mission constituent le *système*. Son état n'est pas directement visible par l'*agent* (le robot). Résoudre un PDMPO revient donc à calculer une stratégie qui remplit la mission au mieux en moyenne, *i.e.* une fonction prescrivant les actions à exécuter selon l'information reçue par l'agent. Ce travail débute par la mise en évidence, dans le contexte robotique, de limites pratiques du modèle PDMPO: elles concernent l'ignorance de l'agent, l'imprécision du modèle d'observation ainsi que la complexité de résolution. Un homologue du modèle PDMPO appelé π-PDMPO, simplifie la représentation de l'incertitude: il vient de la Théorie des Possibilités Qualitatives qui définit la plausibilité des événements de manière qualitative, permettant la modélisation de l'imprécision et de l'ignorance. Une fois les modèles PDMPO et π-PDMPO présentés, une mise à jour du modèle possibiliste est proposée. Ensuite, l'étude des π-PDMPOs factorisés permet de mettre en place un algorithme appelé PPUDD utilisant des Arbres de Décision Algébriques afin de résoudre plus facilement les problèmes structurés. Les stratégies calculées par PPUDD, testées par ailleurs lors de la compétition IPPC 2014, peuvent être plus efficaces que celles des algorithmes probabilistes dans un contexte d'imprécision ou pour certains problèmes à grande dimension. Nous montrons ensuite que les Processus de Markov Cachés possibilistes (π-PMCs), *i.e.* les π-PDMPOs sans les actions, produisent de bons diagnostics dans le contexte de l'intéraction Homme-Machine. Enfin, un PDMPO hybride tirant profit des avantages des modèles probabilistes et possibilistes est présenté: seule la connaissance de l'agent est maintenue sous forme qualitative. Ce modèle mène à une stratégie qui réagit de manière pessimiste au défaut de connaissance, et calculable avec des algorithmes de résolution des Processus Décisionnels de Markov entièrement observables (PDM). Cette thèse propose d'utiliser les possibilités qualitatives dans le but d'obtenir des améliorations en termes de temps de calcul et de modélisation de l'incertitude en pratique.

Mots-clés: PDMPO, Planification dans l'Incertain, Théorie des Possibilités, Robotique Autonome, Connaissance Imprecise