View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by Open Archive Toulouse Archive Ouverte

Open Archive TOULOUSE Archive Ouverte (OATAO)

OATAO is an open access repository that collects the work of Toulouse researchers and makes it freely available over the web where possible.

This is an author-deposited version published in : http://oatao.univ-toulouse.fr/
Eprints ID : 11445

To link to this article :
URL https://zanuttini.users.greyc.fr/jfpda2011/chades.pdf

To cite this version : Ponzoni Carvalho Chanel, Caroline and Teichteil-Königsbuch, Florent and Infantes, Guillaume and Fabiani, Patrick *Modélisation de la faisabilité d'action dans le POMDP avec des préconditions booléennes.* (2011) In: 6èmes Journées Francophones Planification, Décision, et Apprentissage pour la conduite de systèmes, 23 June 2011 - 24 June 2011 (Rouen, France)

Modélisation de la faisabilité d'action dans le POMDP avec des préconditions booléennes

Caroline P. Carvalho Chanel[1,2], Florent Teichteil-Königsbuch[2], Guillaume Infantes[2] and Patrick Fabiani[2]

1. Université de Toulouse - ISAE - Institut Supérieur de l'Aéronautique et de l'Espace
2. Onera - The French Aerospace Lab, F-31055, Toulouse, France, name.surname@onera.fr

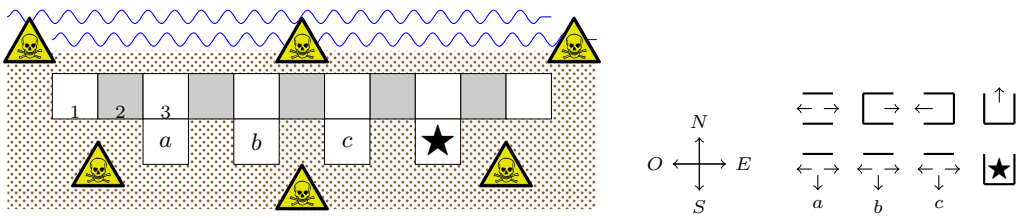
Résumé : En planification classique, une précondition sur une action est une formule booléenne, qui vérifie si une action est réalisable pour un état donné. Cet élément crucial pour des applications réalistes, où par exemple des actions considérées dangereuses doivent être éliminées, n'a pas été formellement modélisé pour les POMDPs à notre connaissance. Une raison est que les préconditions sont définies sur des états, i.e. le domaine d'application de l'action, alors que les décisions prises dans un POMDP sont définies sur l'état de croyance courant de l'agent. Définir simplement des préconditions sur des états de croyance n'est pas suffisant, puisque chaque état de croyance peut-être défini sur plusieurs états, et il n'y a pas de garantie d'éviter que l'agent applique une action infaisable. Augmenter l'espace d'observations avec des actions réalisables n'est pas non plus satisfaisant, d'abord parce que l'information sur les actions applicables est obtenue, par définition, après la décision et, de plus, le processus d'optimisation continuera de maximiser la valeur de l'état de croyance courant sur toutes les actions du modèle. Ainsi, nous proposons une extension du modèle traditionnel des POMDP qui, via une étape additionnelle d'information sémantiquement différente de l'observation standard, permet à l'agent de connaître avec certitude l'ensemble d'actions réalisables avant de décider de la meilleure action à appliquer. Cette étape additionnelle d'information, qui ne nécessite pas de connaître complètement l'état courant de l'agent, requiert une modification significative du modèle de décision, pour lequel nous fournissons un nouveau schéma d'optimisation. Nous comparons la valeur des trajectoires des politiques optimisées pour le modèle traditionnel et pour le modèle proposé, et nous montrons que nos politiques s'avèrent toujours sûres, i.e. sans danger, et expriment donc une valeur plus importante pour des problèmes avec observabilité partielle qui présentent naturellement des préconditions booléennes.

1 Introduction

En planification automatique, les préconditions sont largement utilisées pour modéliser les propriétés de l'environnement indispensables pour la réalisation d'une action. Les préconditions sont des formules booléennes qui représentent le domaine de définition d'une action, i.e. l'ensemble d'états pour lesquels cette action est applicable (Ghallab *et al.*, 2004). Dans de nombreuses applications réalistes, la garantie du respect de ces préconditions est indispensable pour la protection de l'agent robot contre un danger physique réel.

Par exemple, considérons un agent autonome garde-côtes, qui navigue le long des falaises, comme le montre la figure 1(a). Cet exemple est une variation du problème *hallway*, très répandu dans la communauté POMDP, où les murs qui entourent le robot ont été remplacés par des falaises, de sorte que l'agent risque d'y tomber. L'agent peut se retrouver dans un carré, et peut réaliser 4 actions : *nord, sud, est* et *ouest* (figure 1(b)). L'objectif est d'atteindre l'étoile sans tomber d'une falaise de manière *sûre* : pour les états proches d'une falaise, les actions qui peuvent mettre l'agent en péril *doivent* être absolument interdites.

Dans les Processus Décisionnels Markoviens (MDPs), des préconditions booléennes ont été récemment introduites de manière formelle (Younes & Littman, 2003), dans le but de produire des politiques qui contiennent seulement des actions faisables ou désirables dans chaque état atteignable. Les approches précédentes associaient une pénalité grande aux actions indésirables en



(a) environnement côtier (b) actions (c) observations

FIGURE 1 – Problème POMDP d'un robot garde-côtes.

espérant que l'optimisation du processus conduise à proscrire ces actions dans la politique optimale. Cette astuce reste assez hasardeuse car elle n'offre aucune garantie théorique qu'une action infaisable ne sera pas insérée dans la politique optimisée. De plus, ces approches sont aujourd'hui largement dépassées par des algorithmes efficaces qui évaluent directement des préconditions booléennes (Teichteil-Königsbuch *et al.*, 2010).

À notre connaissance, l'utilisation formelle de préconditions n'a jamais été adaptée pour des Processus Décisionnels de Markov *Partiellement Observables* (POMDPs), en dépit des mêmes besoins théoriques et pratiques. La recherche sur les POMDPs reste très focalisée sur les moyens d'améliorer l'efficacité des algorithmes visant à produire une politique pour le modèle POMDP standard, plutôt que sur des améliorations permettant de confronter le modèle standard aux applications réelles. De nombreuses difficultés quant à l'extension du modèle POMDP standard proviennent de l'observabilité partielle : la vérification des préconditions n'est pas directe quand on travaille dans des domaines partiellement observables, car l'état courant n'est pas connu précisément, étant remplacé par une distribution de probabilité sur un ensemble d'états.

La contribution de cet article repose sur la proposition d'un modèle et d'un schéma d'optimisation, compatible avec la majorité des algorithmes développés pour la résolution des POMDPs standards, adapté pour tenir compte convenablement de préconditions booléennes portant sur des paires état-action dans un cadre POMDP. Il nous semble qu'il s'agit de la première approche formelle avec garanties théoriques sur l'exclusion des actions infaisables, permettant de traiter correctement des préconditions booléennes dans un cadre POMDP.

Dans cet article, nous présentons d'abord dans la section 2 le cadre général des POMDP et nous discutons de trois "astuces" permettant d'écarter des actions indésirables, dont l'une est utilisée couramment, en montrant leurs faiblesses et inconvénients. Dans la section 2.2, nous montrons que la solution de contournement la plus utilisée, qui consiste en *choisir la bonne pénalité* pour les paires état-action indésirables, dépend de la pénalité choisie a priori et n'offre aucune garantie que les actions seront effectivement exclues. Dans la section 2.3, nous présentons l'option de définir des préconditions sur des états de croyance ; nous montrons que lier des préconditions à des états de croyance nous ramène à des politiques incohérentes. Dans la section 2.4, nous présentons une troisième option qui consiste en augmenter l'espace d'observations avec les ensembles des actions réalisables, certes plus réaliste mais qui n'offre toujours pas de garantie sur l'exclusion de paires état-action indésirables. Notre contribution, présentée dans la section 3, se base sur cette troisième approche : nous proposons de séparer l'étape (standard) d'observation de l'état et celle (nouvelle) d'observation des actions réalisables de sorte à "forcer" une mise à jour supplémentaire de l'état de croyance qui garantisse d'appliquer une action désirable à l'étape de décision courante. Dans la section 4, nous montrons que notre nouveau modèle est compatible avec les différents algorithmes pour les POMDPs standards. Et finalement, nous présentons et nous discutons nos résultats comparatifs préliminaires qui démontrent l'intérêt de modéliser formellement les préconditions dans un cadre POMDP pour des problèmes où certaines actions doivent être écartées avec certitude de la politique optimisée préalablement à l'optimisation du problème.

2 Contexte

2.1 POMDP — aperçu général

Formellement, un POMDP est défini comme un n-uplet $\langle S, A, \Omega, T, O, R, b_0 \rangle$ où : S est un ensemble d'états, A un ensemble d'actions, Ω un ensemble d'observations. Pour tout instant de décision $t \in \mathbb{N}$, $T : S \times A \times S \to [0; 1]$ est une fonction de transition entre les états où $\forall a \in A, \forall s_t \in S$, et $\forall s_{t+1} \in S$ on

définit : $T(s_t, a, s_{t+1}) = p(s_{t+1} \mid a, s_t)$; $O : \Omega \times S \to [0; 1]$ une fonction d'observation où $\forall o_t \in \Omega, \forall a \in A$, et $\forall s_t \in S$, on définit : $O(o_t, s_t) = p(o_t|s_t, a)$; $R : S \times A \to \mathbb{R}$ une fonction de récompenses associées aux couples état-action ; et b_0 une distribution de probabilité sur les états initiaux, appelée *état de croyance*. On note Δ l'ensemble des distributions de probabilités sur les états, appelé aussi espace (continu) d'états de croyance.

À chaque pas de temps t, l'agent choisit une action $a \in A$ étant donné l'état de croyance $b_t \in \Delta$, qui l'amène stochastiquement à un nouvel état $s_{t+1} \in S$; ensuite, l'agent perçoit une observation bruitée $o \in \Omega$. L'agent met à jour son *état de croyance*, grâce à la règle de Bayes, ce qui dépend de l'action réalisée, de l'observation reçue, et est fonction de l'état de croyance précédent ($s' = s_{t+1}$ suit l'état $s = s_t$).

$$b_a^o(s') = \frac{p(o|s') \sum_{s \in S} p(s'|s, a) b(s)}{\sum_{s \in S} \sum_{s'' \in S} p(o|s'') p(s''|s, a) b(s)} \tag{1}$$

Afin de simplifier la lecture, nous noterons $b_t(s) = Pr(s_t = s)$.

L'objectif de la résolution d'un POMDP est de construire une politique, c'est-à-dire une fonction $\pi : \Delta \to A$, qui maximise un critère de performance. L'espérance de la somme pondérée pour tout état de croyance initial $V^\pi(b) = E_\pi \left[\sum_{t=0}^{\infty} \gamma^t r(b_t, \pi(b_t)) \mid b_0 = b \right]$ est généralement optimisée. La valeur de la politique optimale π^* est définie par la fonction de valeur optimale qui satisfait l'équation d'optimalité de Bellman :

$$V^*(b) = \max_{a \in A} \left[\sum_{s \in S} r(s, a) b(s) + \gamma \sum_{o \in O} p(o|a, b) V^*(b_a^o) \right] \tag{2}$$

Cette fonction de valeur est linéaire par morceaux et convexe (Smallwood & Sondik, 1973), i.e, à l'instant $n \leq \infty$, la fonction de valeur V_n peut être représentée par des hyperplans sur Δ, nommés α-vecteurs. Un α-vecteur et l'action associée $a(\alpha_n^i)$ définissent une région dans l'espace de croyance pour lequel ce vecteur maximise V_n. Donc, la valeur d'un état de croyance peut être définie comme $V_n(b) = \max_{\alpha_n^i \in V_n} b \cdot \alpha_n^i$. Et la politique optimale à cette étape est $\pi_n(b) = a(\alpha_n^b)$.

2.2 Introduire des pénalités pour inhiber certaines actions dans certains états

Supposons que l'état de croyance est différent de zéro seulement pour les états $1, 2$ et 3 de la figure 1(a). Une solution simple, et largement utilisée, consiste à associer un coût très important (idéalement $+\infty$) pour les actions infaisables : *nord*, *ouest* et *sud* dans l'état 1, *nord* et *sud* dans l'état 2, et *nord* pour l'état 3. Comme des valeurs infinies ne peuvent pas être modélisées proprement par des librairies informatiques, nous devons "régler" une valeur finie *suffisamment grande* à la place de $+\infty$. Néanmoins, le seuil de cette valeur qui garantit d'inhiber des actions infaisables en optimisant la politique dépend en fait de la valeur optimale des états inconnue a priori ; cette pénalité est fonction des récompenses intrinsèques au modèle et du critère d'optimisation utilisé. En d'autres termes, pour un critère d'optimisation donné, nous devrions résoudre le problème plusieurs fois *avant* de connaître le seuil correct pour chaque pénalité qui sera associée aux actions infaisables. Par exemple, si atteindre l'étoile donne à l'agent la récompense de 1, et toutes les actions sans danger n'ont pas de coût, alors pour un $\gamma = 0.9$, même avec un coût de 50, des actions infaisables sont encore exécutées 139 fois sur 500 simulations avec un horizon de 50. Une autre alternative peut consister en redéfinir toutes les opérations algébriques et informatiques de bas niveau afin de traiter correctement des valeurs réellement infinies, mais ceci est un travail fastidieux qui peut de plus induire de nombreuses erreurs de calculs complexes d'optimisation.

2.3 Préconditions basées sur l'état de croyance

Intuitivement, une façon a priori correcte de modéliser des actions infaisables consiste à définir des préconditions sur des états de croyance. Or, la notion de précondition repose sur des états réels du monde, et non sur des états de croyance : nous avons donc besoin d'adapter cette notion aux états de croyance.

Une approche pessimiste pourrait consister à interdire une action dès qu'il existe une probabilité non nulle de conduire l'agent à un danger imminent, mais cela pourrait interdire une action utile :

dans notre exemple, si l'état de croyance n'est pas nul sur les états 1, 2 et 3 de la figure 1(a), la seule action sûre (qui ne conduit l'agent avec certitude à aucune falaise) est *est*, qui par chance conduit l'agent vers l'étoile. Mais, si l'étoile était au contraire dans le carré a, l'action *est*, qui est la seule action sûre et autorisée en considérant un état de croyance non nul sur les états 1, 2 et 3, ne conduit plus au but, Ainsi, nous écartons complètement cette approche qui s'avère trop simpliste en pratique.

Par contre, une autre approche optimiste serait d'élaguer au minimum les actions infaisables. Une stratégie simple consisterait à calculer un ensemble d'actions réalisables basé sur les états supports de l'état de croyance b (états pour lesquels la probabilité $b(s) \neq 0$). Une telle information n'empêche cependant pas de manière sûre l'agent d'appliquer des actions dangereuses. Considérons une fois de plus notre exemple du problème du robot garde-côtes (figure 1(a)) et un état de croyance non nul sur les états correspondants aux numéros 1, 2 et 3. Suivant cette définition, l'ensemble des actions réalisables est $\{ouest, est, sud\}$ pour cet état de croyance, ce qui est problématique puisque nous savons que pour l'état 1 l'agent ne devrait appliquer que l'action *est* pour être hors de danger. Le choix de l'action optimale dans cet état de croyance pourrait conduire l'agent à appliquer une décision dangereuse. Plus généralement, si l'état de croyance est non nul sur des états où l'ensemble des actions réalisables est mutuellement exclusif, cette stratégie simple produira des politiques incohérentes qui appliqueront des actions dangereuses à l'exécution.

2.4 Augmenter le modèle d'observation avec des ensembles d'actions autorisées

Puisqu'aucune des approches présentées précédemment n'est complètement satisfaisante, une autre approche intuitive est de rajouter à l'espace des observations les ensembles des actions réalisables, modélisé comme une variable additionnel d'observation (figure 1(c)). Cette information additionnelle devrait forcer le robot à ne choisir que des actions sûres lors de l'exécution, mais là-aussi une étude plus approndie montre les écueils de cette approche.

D'abord, considérons un état de croyance initial b_0 : s'il est uniforme, comme il est coutume de faire, il n'y a aucun moyen d'interdire une action dangereuse à la première étape, ce qui signifie qu'une observation initiale est requise, reliée à aucune transition, ou que l'état de croyance initial doit être consistent avec l'ensemble des actions sûres. À cette fin, l'état de croyance initial doit être donné "à la main", ce qui est souvent source d'erreurs ; en particulier, si b_0 attribue une probabilité nulle à de nombreux états, alors la mise à jour de la croyance pourrait induire des incohérences à chaque nouvelle observation.

Ensuite, considérons un état de croyance non nul seulement sur les états 1, 2 et 3. Les observations espérées sont $\boxed{\rightarrow}$, $\overleftrightarrow{\square}$, $\overleftrightarrow{\downarrow_a}$ et $\boxed{\uparrow}$. En regardant les actions contenues dans ces observations, l'équation 2 sera optimisée pour toutes les actions du modèle ($\max_{a \in A}$). Par conséquent, même si l'observation $\overleftrightarrow{\square}$ est reçue, l'action optimisée, qui est définie sur la base des quatre observations possibles, pourra être *sud*, en menant peut-être l'agent vers une falaise. Aussi, nous pensons que l'information concernant l'action doit être fournie *avant* la décision et non après. Pour cela, l'opérateur d'agrégation, ici $\max_A$, doit être redéfini, en particulier pour filtrer certaines actions de l'ensemble A.

Plus précisément, l'ensemble d'actions réalisables doit être construit plus intelligemment, en ajoutant une étape supplémentaire d'information, *a priori* très similaire à l'observation standard, mais sémantiquement différente, qui informe l'agent de l'ensemble courant des actions réalisables indépendamment de son état de croyance courant. En effet, les observations dépendent des actions réalisées, donc ces deux informations doivent être clairement séparées.

3 Optimisation de POMDP avec des préconditions

Comme discuté plus haut, la seule façon de prendre en compte directement des préconditions booléennes dans les POMDP (par opposition à des façons indirectes comme le réglage de coûts où l'inclusion des actions réalisables dans les observations) est de rajouter une étape d'information supplémentaire, afin de restreindre l'état de croyance à des états qui ont le même ensemble d'actions réalisables, en redéfinissant l'opérateur d'agrégation et en optimisant la politique seulement avec

des actions réalisables. Ceci nous ramène à une adaptation importante de la mise à jour de l'état de croyance et du processus d'optimisation des POMDP, ainsi que nous le détaillons dans cette section. Avant de présenter ces changements, nous introduisons les fondements de notre approche, avec les définitions et notations correspondantes.

3.1 Information sur la faisabilité d'une action

Une précondition est une formule de valeur booléenne (ou litéral) qui doit être vraie *sûrement* si et seulement si une action est applicable dans un état donné. Nous notons $\mathcal{A}_f(s)$ l'ensemble des actions réalisables dans un état s. Une précondition basée sur un état est définie par une relation de faisabilité $\mathbb{I}$ qui indique si pour un état s, une action a est applicable ou non : $\mathbb{I}(a,s) = \mathbf{1}_{a \in \mathcal{A}_f(s)}$ où, $\mathbf{1}_{cond}$ est 1 si *cond* est vrai, ou 0 sinon. $\mathbb{I}(a,s)$ peut être aussi vu comme la probabilité 1 ou 0 de l'applicabilité d'une action a sachant l'état s, i.e., $\mathbb{I}(a,s) = Pr(a \in \mathcal{A}_f(s) | s_t = s)$.

Étape additionnelle d'information

Ici, nous supposons que l'agent reçoit une information supplémentaire de l'environnement pendant l'exécution de la politique, entre l'observation standard et l'étape de décision, qui consiste en l'ensemble des actions réalisables. Cette procédure est souvent utilisée dans les systèmes autonomes au travers de fonctionnalités spécifiques et découplées de la planification, comme le contrôle d'exécution dans (Ingrand *et al.*, 2007), dans le but d'éviter des dommages que pourrait subir le robot. Généralement, l'exécution de la politique doit être contrôlée pour garantir que le plan sera réalisé en toute sécurité.

De manière à prendre compte des contraintes de sécurité dans l'optimisation et dans l'exécution de la politique, les modèles POMDP doivent être étendus. Comme nous ne connaissons pas à l'avance l'ensemble des actions réalisables que sera reçu de l'environnement, nous devons planifier pour *tous les ensembles d'actions réalisables possibles* $\{\widetilde{\mathcal{A}_f}^1, \cdots, \widetilde{\mathcal{A}_f}^j\}$, *indépendamment de l'état de croyance de l'agent*, avec $j = 2^{|A|} - 1$ combinaisons d'actions différentes, où $|A|$ est le nombre total d'actions du modèle. Pour notre exemple du robot garde-côtes (Fig.1(a)), des combinaisons d'actions possibles sont : $\{est, ouest\}$, $\{est, sud\}$, $\{est, ouest, sud\}$, $\{est, sud\}$ et $\{ouest\}$.

Le point clef est le suivant : si nous disposons de l'information sur l'ensemble des actions réalisables, la fonction indicative conjointe pourrait être utilisée dans une mise à jour supplémentaire de l'état de croyance avant l'étape de décision. Ceci nous permettrait d'optimiser une fonction de valeur sur un état de croyance qui serait distribué seulement sur des états qui possèdent le même ensemble d'actions réalisables. Aussi, nous pourrions optimiser la valeur de chaque état seulement sur l'ensemble des actions faisables et les observations correspondantes. En d'autres termes, comme les observations dépendent des actions réalisées, l'étape d'observation et l'étape d'information sur les actions réalisables doivent être séparées, ce qui signifie que nous avons deux étapes informatives distinctes.

Nous définissons la fonction indicative conjointe d'un ensemble d'actions $\mathcal{U} \subset A$ pour un état s comme :

$$\mathbb{I}(\mathcal{U}, s) = \prod_{a_i \in \mathcal{U}} \mathbb{I}(a_i, s) \prod_{a_j \notin \mathcal{U}} (1 - \mathbb{I}(a_j, s)) \tag{3}$$

Nous avons directement $\mathbb{I}(\mathcal{A}_f(s), s) = 1$. De plus, il est intéressant de remarquer que $\mathbb{I}(\mathcal{U}, s) = Pr(\mathcal{A}_f(s) = \mathcal{U} | s)$, est la probabilité 1 ou 0 que l'ensemble des actions conditionné à l'état s soit égal à l'ensemble des actions réalisables $\mathcal{A}_f(s)$. Cette relation nous permettra de mettre à jour l'état de croyance en fonction de l'ensemble courant des actions réalisables. Par exemple, en considérant la figure 1(a), pour un état de croyance uniformément distribué b, l'agent reçoit $\mathcal{A}_f = \{west, east\}$ en tant qu'ensemble d'actions réalisables, puis il projette son état de croyance conditionné à $\mathcal{A}_f$ et obtient $\tilde{b}_{\mathcal{A}_f}$. Nous pouvons vérifier que l'incertitude sera maintenant répartie sur les états de couleur grise dans la figure 1(a), pour lesquels l'ensemble d'actions réalisables est le même. Il convient de noter que l'état de croyance n'est pas systématiquement réduit à un seul état après l'information sur les actions réalisables, même en gardant le caractère d'observabilité partielle de l'agent.

Utilisation de l'information sur les états supports

Il est possible d'inclure dans notre approche une technique pour élaguer au minimum les actions réalisables (section 2.3) en se basant sur l'état de croyance. Toutefois, cette approche pourrait induire une incohérence si l'état de croyance est nul sur le vrai état caché du système.

D'abord, nous définissons l'ensemble des actions réalisables connaissant l'état de croyance b, comme :

$$\mathcal{A}_f(b) = \{a \in A | \exists s \in S, b(s) \neq 0 \wedge \mathbb{I}(a,s) = 1\} \tag{4}$$

Maintenant, supposons que le robot reçoit l'ensemble d'actions réalisables $\widetilde{\mathcal{A}_f{}^i}$ à l'instant t. Avec la définition 4, nous pouvons restreindre l'ensemble d'actions réalisables à $\mathcal{A}_f^i = \mathcal{A}_f(b) \cap \widetilde{\mathcal{A}_f{}^i}$. Le schéma d'optimisation correspondant signifie que nous optimisons la valeur pour toutes les intersections, telles que plusieurs d'entre elles sont heureusement vides, ou résultent en de petits ensembles comparés à l'approche qui n'utilise pas d'information sur les états supports de b. Nous notons $\{\mathcal{A}_f^1, \cdots, \mathcal{A}_f^n\}$ l'ensemble des ensembles d'actions réalisables, i.e., toutes les combinaisons d'actions intersectées avec $\mathcal{A}_f(b)$. Nous avons $n = 2^{|\mathcal{A}_f(b)|} - 1$. Sans perte de généralité, si l'information supplémentaire n'est pas disponible, l'intersection $\mathcal{A}_f(b) \cap \widetilde{\mathcal{A}_f{}^i}$ n'est pas définie et $\mathcal{A}_f^i = \mathcal{A}_f(b)$.

Trois cas peuvent nous ramener à une intersection vide. Premièrement, $\mathcal{A}_f(b) = \emptyset$ est impossible, car $\sum_s b(s) = 1$. Deuxièmement, $\widetilde{\mathcal{A}_f{}^i} = \emptyset$ est aussi impossible, car il existe au moins une action par ensemble d'actions quand les combinaisons sont générées. Troisièmement, si $\mathcal{A}_f(b)$ et $\widetilde{\mathcal{A}_f{}^i}$ sont mutuellement exclusifs, alors il y a incohérence entre l'état de croyance et l'information externe ; nous ne prenons donc pas non plus en compte ce cas.

3.2 Mise à jour de l'état de croyance conditionnée à l'ensemble des actions réalisables

Cette nouvelle information disponible pour l'agent nous ramène à une mise à jour en deux étapes de l'état de croyance de l'agent.

1. Une projection de b est faite pour un ensemble d'actions réalisables $\mathcal{A}_f^i \in \{\widetilde{\mathcal{A}_f{}^1}, \cdots, \widetilde{\mathcal{A}_f{}^j}\}$(ou $\mathcal{A}_f^i \in \{\mathcal{A}_f^1, \cdots, \mathcal{A}_f^n\}$ suivant l'approche utilisée), résultant en $\tilde{b}_{\mathcal{A}_f^i}$.

$$\begin{aligned} \tilde{b}_t(s)_{\mathcal{A}_f^i} &= Pr(s_t = s|\mathcal{A}_f^i) = \frac{Pr(\mathcal{A}_f^i|s_t = s)Pr(s_t = s)}{Pr(\mathcal{A}_f^i)} \\ \tilde{b}_t(s)_{\mathcal{A}_f^i} &= \frac{\mathbb{I}(\mathcal{A}_f|s_t = s)b_t(s)}{\sum_{s''} \mathbb{I}(\mathcal{A}_f|s_t = s'')b_t(s'')} \end{aligned} \tag{5}$$

   Si l'état de croyance est incohérent avec l'ensemble des actions réalisables, le dénominateur devient nul, ce qui invalide la règle de Bayes. Ceci peut arriver si $\mathcal{A}_f^i = \widetilde{\mathcal{A}_f{}^i}$, i.e., quand $\forall a \in \widetilde{\mathcal{A}_f{}^i}$, $a \notin \mathcal{A}_f(b)$. Dans ce cas, nous pouvons écarter l'état de croyance courant et simplement le réinitialiser.

2. Une action $a \in \mathcal{A}_f^i$ et une observation $o \in \Omega$ ramènent l'agent à un état de croyance b_a^o. Le reste de la mise à jour de $\tilde{b}$ se fait de manière identique au modèle classique avec la contrainte supplémentaire que l'action choisie appartient à $\mathcal{A}_f$, et que b est substitué par $\tilde{b}$ dans l'équation 1 :

$$b^o_{a \in \mathcal{A}_f}(s') = \frac{p(o|s') \sum_{s \in S} p(s'|s,a)\tilde{b}(s)}{\sum_{s \in S} \sum_{s'' \in S} p(o|s'')p(s''|s,a)\tilde{b}(s)} \tag{6}$$

L'étape supplémentaire d'information n'invalide pas l'hypothèse que l'état de croyance est un état complet d'information, car le processus global reste Markovien. De plus, à l'exécution du plan, l'agent devra prendre des décisions basées sur $\tilde{b}$; par exemple aux étapes k et $k+1$ de la figure 2, après l'étape informative sur l'ensemble d'actions réalisables. Ainsi, l'algorithme d'optimisation devra implémenter des mises à jour de la valeur (des *backups*) sur les $\tilde{b}$. Dans les cas où l'information sur les actions réalisables n'est pas disponible, le schéma est réduit au schéma standard, à condition que $a \in \mathcal{A}_f(b)$.

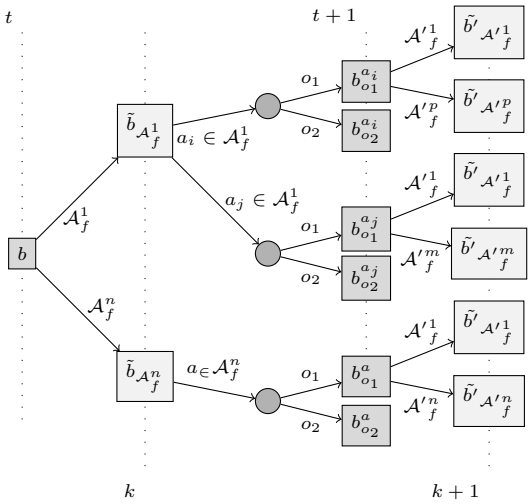


FIGURE 2 – Nouveau schéma de mise à jour de l'état de croyance.

3.3 Optimisation : extension de l'opérateur de *backup*

Puisque nous ajoutons une étape supplémentaire d'information au modèle de mise à jour, nous devons aussi adapter l'équation de Bellman pour calculer la valeur espérée d'un état de croyance $\tilde{b}$ en fonction des états de croyance intermédiaires $b^o_{a\in\mathcal{A}_f}$:

$$V_{k+1}(\tilde{b}_{\mathcal{A}_f}) = \max_{a\in\mathcal{A}_f}\{r(\tilde{b},a) + \sum_o p(o|a,\tilde{b})V_n(b^o_{a\in\mathcal{A}_f})\} \tag{7}$$

où $\mathcal{A}_f$ représente l'ensemble des actions réalisables associé à l'état de croyance $\tilde{b}$. Notons que la valeur $V_{k+1}(\tilde{b}_{\mathcal{A}_f})$ est linéaire en $\tilde{b}$, mais elle dépend aussi de la valeur des futurs $b^o_{a\in\mathcal{A}_f}$. Le calcul de $V_n(b^o_{a\in\mathcal{A}_f})$ n'est pas trivial, car il dépend du $\tilde{b}_{\mathcal{A'}_f}$suivant. Nous calculons $V_n(b^o_{a\in\mathcal{A}_f})$ sous forme d'une moyenne :

$$V_n(b) = \sum_{i=1}^{|C(b)|} Pr(\mathcal{A}_f^i)V_k(\tilde{b}_{\mathcal{A}_f^i}(b)) \tag{8}$$

Ceci ramène à une dépendance non linéaire, car le nombre de successeurs de b, noté $|C(b)|$, dépend du nombre d'ensembles possibles d'actions réalisables. La valeur moyenne des successeurs de b est valide seulement localement : en d'autres termes, nous connaissons le nombre de successeurs seulement pour un b donné. Par conséquent, nous pouvons seulement approximer la valeur localement avec un α-vecteur. Toutefois, en tenant compte de cette remarque et en utilisant le fait que l'équation 7 est linéaire en $\tilde{b}$, nous redéfinissons l'opérateur de *backup* présenté en (Pineau *et al.*, 2003). Ainsi, $\forall a \in \mathcal{A}_f$, $\forall o \in \Omega$ et $\forall\bar{\alpha}_i \in V_n$, avec V_n défini par l'équation 8 :

$$\begin{aligned}\Gamma^{a,*} &\leftarrow \alpha^{a,*}(s) = R(s,a)\mathbb{I}(a,s)\\ \Gamma^{a,o} &\leftarrow \alpha^{a,o}(s) = \gamma\sum_{s'\in S} p(s'|s,a)\mathbb{I}(a,s)p(o|s')\bar{\alpha}_i(s')\end{aligned}$$

L'α-vecteur associé à $a \in \mathcal{A}_f$ est construit de la manière suivante :

$$\Gamma^a_{\tilde{b}} = \Gamma^{a,*} + \sum_{o\in\Omega}\arg\max_{\alpha\in\Gamma^{a,o}}(\alpha\cdot\tilde{b}) \tag{9}$$

Nous utilisons l'α-vecteur qui maximise $\tilde{b}$ pour calculer V_{k+1} :

$$V_{k+1} \leftarrow \underset{\Gamma^a_{\tilde{b}},\forall a\in\mathcal{A}_f}{\arg\max}\,(\Gamma^a_{\tilde{b}}\cdot\tilde{b}) \tag{10}$$

Comme V_k peut être représenté par des α-vecteurs, l'équation 8 devient :

$$V_n(b) = \sum_{i=1}^{|C(b)|} Pr(\mathcal{A}_f^i)\arg\max_{\tilde{\alpha}\in V_k}\tilde{\alpha}\cdot\tilde{b}(b) \tag{11}$$

La relation entre $\tilde{b}$ et b est donnée par l'équation 5, qui peut s'exprimer sous forme vectorielle par $\tilde{b} = \frac{\mathbb{I}(\mathcal{A}_f)b}{Pr(\mathcal{A}_f)}$, où $\mathbb{I}(\mathcal{A}_f)$ est la matrice diagonale $S \times S$ dont les termes diagonaux sont $\mathbb{I}_{ii} = \mathbb{I}(\mathcal{A}_f, s_i)$. Ainsi :

$$V_n(b) = \sum_{i=1}^{|C(b)|} Pr(\mathcal{A}_f^i) \arg\max_{\tilde{\alpha} \in V_k} \tilde{\alpha} \cdot \frac{\mathbb{I}(\mathcal{A}_f^i)b}{Pr(\mathcal{A}_f^i)} \quad (12)$$

$$V_n(b) = \sum_{i=1}^{|C(b)|} \arg\max_{\tilde{\alpha} \in V_k} \tilde{\alpha} \cdot \mathbb{I}(\mathcal{A}_f^i)b \quad (13)$$

Finalement, V_n est obtenu de la manière suivante :

$$V_n \leftarrow \sum_{i=1}^{|C(b)|} \arg\max_{\tilde{\alpha} \in V_k} \tilde{\alpha} \cdot \mathbb{I}(\mathcal{A}_f^i) \quad (14)$$

$\mathbb{I}(\mathcal{A}_f^i)$ opère comme un masque sur b, en tenant compte dans la construction de l'α-vecteur des valeurs définies seulement pour les états où $\mathcal{A}_f^i$ est l'ensemble des actions réalisables. La valeur de b est localement approchée en moyennant les α-vecteurs de ses successeurs. Si l'information sur l'ensemble des actions réalisables n'est pas disponible, le nouvel opérateur se résume à l'opérateur standard (Pineau *et al.*, 2003), mais les projections et le calcul des α-vecteurs seront restreints aux actions a telles que $a \in \mathcal{A}_f(b)$.

Quelques avantages de ce nouvel opérateur de *backup* peuvent être évoqués : (1) En tenant compte des actions qui appartiennent à l'ensemble des actions réalisables pour chaque état de croyance $\tilde{b}$, moins d'évaluations d'actions sont réalisées par rapport au modèle standard. (2) Les nouveaux α-vecteurs sont creux, ce qui peut être exploité dans le calcul de la valeur. Cette faible densité est due à l'(in-)faisabilité d'une action dans un état donné ; les α-vecteurs sont des hyperplans définis sur l'espace des états de croyance, donc si pour un état s une action est interdite, la valeur associée à cette action n'est pas définie.

Théorème 1 (Contraction du nouvel opérateur de *backup*)
Soit $\gamma < 1$. Le nouvel opérateur de backup *est défini comme :*

$$\mathcal{L}V(\tilde{b}) = \max_{a \in \mathcal{A}_f} r(\tilde{b}, a) + \gamma \sum_{o \in \Omega} p(o|a, \tilde{b}) \sum_{i=1}^{C(b_a^o)} Pr(\mathcal{A}_f^i) V(\tilde{b}_{\mathcal{A}_f^i}^{a,o})$$

est une contraction sur $\mathcal{V}$, l'espace des fonctions de valeur.

Preuve Soit $V \in \mathcal{V}$, $U \in \mathcal{V}$, et $\tilde{b} \in \Delta$, où $||\tilde{b}|| = \sum_{s \in S} |\tilde{b}(s)| = 1$. Nous supposons que $\mathcal{L}V(\tilde{b}) \geq \mathcal{L}U(\tilde{b})$, et

$$a^* = \arg\max_{a \in \mathcal{A}_f} \{r(\tilde{b}, a) + \gamma \sum_{o \in \Omega} p(o|a, \tilde{b}) \sum_{i=1}^{C(b_a^o)} Pr(\mathcal{A}_f^i) V(\tilde{b}_{\mathcal{A}_f^i}^{a,o})\}$$

Nous avons :

$$\begin{aligned} |\mathcal{L}V(\tilde{b}) - \mathcal{L}U(\tilde{b})| = \mathcal{L}V(\tilde{b}) - \mathcal{L}U(\tilde{b}) &\leq \gamma \sum_{o \in \Omega} p(o|a^*, \tilde{b}) \sum_{i=1}^{C(b_{a^*}^o)} Pr(\mathcal{A}_f^i) ||V - U|| \\ &\leq \gamma ||V - U|| \end{aligned}$$

donc : $||\mathcal{L}V - \mathcal{L}U|| = \max_{||\tilde{b}||=1} |\mathcal{L}V(\tilde{b}) - \mathcal{L}U(\tilde{b})| \leq \gamma ||V - U||$ □

3.4 Discussion sur la complexité

Afin d'étudier la complexité de calcul de notre modèle étendu de POMDP, on voit que tous les ensembles possibles des actions réalisables doivent être évalués à chaque pas de temps, donc, dans le pire cas, la complexité de l'algorithme POMDP est multiplié par $2^{|A|} - 1$, où "-1" représente l'ensemble vide. Ceci est le prix à payer pour garantir qu'aucune action infaisable ne sera appliquée lors de l'exécution de la politique. Cependant ceci est le pire cas, puisque dans la pratique, comme montré dans nos résultats expérimentaux (voir la prochaine section), l'état de croyance est généralement bien élagué avant l'étape de décision, ce qui réduit le nombre d'α-vecteurs générés.

Nous soulignons que nous présentons un nouveau modèle dont le but est d'éviter que des actions infaisables soient choisies pour l'exécution, ce qui n'est pas formellement garanti par le modèle POMDP standard. De plus, notre intention dans la section suivante n'est pas forcément de comparer l'efficacité des algorithmes, parce que, comme remarqué précédemment, cette garantie a forcément un coût de calcul.

4 Résultats expérimentaux

Les algorithmes récents de résolution POMDPs, comme PBVI (Pineau *et al.*, 2003), Perseus (Spaan & Vlassis, 2004), HSVI2 (Smith & Simmons, 2005), SARSOP (Kurniawati *et al.*, 2008) approximent la fonction de valeur en utilisant un ensemble fini B d'états de croyance, avec $B \subset \Delta$. Ces algorithmes implémentent différentes heuristiques pour explorer l'espace d'états de croyance et pour mettre à jour la fonction valeur V, définie comme un ensemble d'α-vecteurs, pour chaque b exploré. Donc, V doit contenir au maximum $|B|$ α-vecteurs. Leur principale différence est la façon dont ils parcourent l'espace d'états de croyance, ou la façon dont ils maintiennent une borne supérieure ou inférieure sur la fonction de valeur.

4.1 Protocole expérimental

Dans le but de valider notre approche, nous avons implémenté PCVI – *PreCondition Value Iteration*, un algorithme *point-based* qui s'appuie sur PBVI. PCVI travaille sur un ensemble fini $\tilde{B} = \tilde{b}_0, ...\tilde{b}_k$ d'états de croyance et utilise la nouvelle mise-à-jour de l'état de croyance et le nouvel opérateur de *backup* nous permettant de prendre en compte les préconditions, conditionnées ou non à une information extérieure. PCVI, comme PBVI et Perseus, explore l'espace d'états de croyance par des trajectoires stochastiques, et cette exploration dépend de la disponibilité ou non de l'information supplémentaire sur les actions réalisables. Les deux approches présentées dans la section 3 peuvent être choisies. Sans perte de généralité, si aucune information additionnelle n'est disponible, $\mathcal{A}_f$ est défini comme dans l'approche d'élagage minimal basé sur les états support de b présenté dans la section 2.3, équation 4. Nous soulignons que ce schéma de décision peut être appliqué à n'importe quel algorithme POMDP "standard".

Ici, nous comparons des politiques optimisées via le modèle standard du POMDP avec des politiques optimisées via notre étape additionnelle d'information sur les actions réalisables, en termes de réussite de mission et de sécurité. Même si nous ne sommes pas forcément intéressés par la comparaison des performances des algorithmes mais par la comparaison des modélisations, nous remarquons que notre étape additionnelle d'information n'augmente pas le temps de calcul et ne diminue pas les récompenses reçues.

Nous avons réalisé des expériences dans 8 problèmes POMDP très connus de la communauté : *maze4x3*, *maze4x5x2*, *Hallway*, *Hallway2*, *aircraft* et *iff* (Cassandra, 1998), *tiger-grid* et *RockSample4x4* (Smith & Simmons, 2005), avec les algorithmes PBVI, HSVI2, et PCVI. Pour les problèmes de navigation, il est interdit à l'agent de se mouvoir vers un mur. Dans rockSample, il est interdit de ramasser des pierres là où il n'y en a pas. Dans les problèmes *aircraft*, il est interdit d'utiliser le radar actif si une cible est trop proche de la base.

Dans les algorithmes classiques, comme PBVI et HSVI2, la faisabilité d'une action n'est pas formellement prise en compte : nous avons donc associé un grand coût aux paires état-action (s, a) infaisables afin d'éviter que la politique ne fasse exécuter de telles actions dans les états concernés. Même si nous avions modélisé l'information additionnelle sur l'ensemble des actions réalisables par une variable d'observation supplémentaire, ces algorithmes ne pourraient pas traiter correctement cette information (voir section 2.4).

4.2 Résultats comparatifs

Dans le tableau 1 sont résumées les performances moyennes sur 500 simulations. On voit que pour la majorité des problèmes, les récompenses reçues pour PCVI et PCVI+ sont plus importantes. Ceci s'explique par le fait que notre algorithme ne réalise aucune action infaisable et donc n'est pas "puni" avec une pénalité (nous calculons directement une politique que garantit qu'aucune action infaisable ne sera choisie).

La figure 3 montre plus de résultats pour les problèmes *maze4x3* et *hallway[1,2]*. On compare notre modèle avec le modèle standard sur trois critères : (1) le nombre de fois qu'une action infaisable est choisie ; (2) le pourcentage de réussite de mission ; et (3) le nombre d'étapes pour atteindre le but. Ces critères ont été mesurés pour 500 exécutions de la politique. Le critère (1) est le principal de notre étude. Les critères (2) et (3) sont présentés pour souligner que notre modèle étendu est compétitif avec le modèle classique du POMDP, sinon meilleur, pour ces critères. Pour

	maze4x3 (11s/4a/6o) p.1 h.50			**maze4x5x2** (39s/4a/4o) p.1 h.100		
	R	Temps(s)	$\|\Gamma\|$	R	Temps(s)	$\|\Gamma\|$
PBVI	0.089	5.913	25	0.061	191.86	166
HSVI2	0.015	65.25	260	-0.077	14.29	1132
PCVI-	0.072	8.344	27	-0.064	373.4	201
PCVI	0.540	0.880	3	0.634	43.60	24
PCVI+	0.552	2.757	7	0.470	53.77	23
	hallway (60s/5a/21o) p.1 h.250			**hallway2** (92s/5a/17o) p.1 h.250		
	R	Temps(s)	$\|\Gamma\|$	R	Temps(s)	$\|\Gamma\|$
PBVI	0.470	1747	251	0.204	4489	491
HSVI2	0.480	2729	935	0.258	313.3	2047
PCVI-	0.482	3140	273	0.202	6236	482
PCVI	0.516	1305	259	0.310	973.7	278
PCVI+	0.541	1670	262	0.344	977.5	291
	aircraft (12s/6a/5o) p.1 h.50			**iff** (104s/4a/22o) p.30 h.100		
	R	Temps(s)	$\|\Gamma\|$	R	Temps(s)	$\|\Gamma\|$
PBVI	12.84	114.2	26	7.262	5839	420
HSVI2	13.26	0.065	3	-7.163	2839	13983
PCVI-	12.13	142.8	23	7.746	10322	421
PCVI	14.30	129.4	35	7.257	4093	244
PCVI+	15.02	120.3	22	8.425	7480	344
	tiger-grid (36s/5a/17o) p.10			**rockSample4x4** (257s/9a/2o) p.100		
	R	Temps(s)	$\|\Gamma\|$	R	Temps(s)	$\|\Gamma\|$
PBVI	0.579	2343	418	16.29	15893	65
HSVI2	0.532	5432	14168	18.15	1.059	287
PCVI-	0.617	2295	399	16.77	5683	111
PCVI	0.625	1046	286	16.36	8423	94
PCVI+	0.632	841.6	220	15.81	26922	76

R = Récompenses, p. = penalité, h. = horizon, $|\Gamma|$ = nombre de α-vecteurs, PCVI- = états-support, PCVI = information additionnelle, PCVI+ = mélange (intersection) des deux autres.

TABLE 1 – Comparaison entre les différents algorithmes.

le problème *maze4x3*, même pour une pénalité de 50, des actions infaisables sont encore appliquées avec le modèle standard. Dans les problèmes *hallway*, l'utilisation des actions infaisables diminue

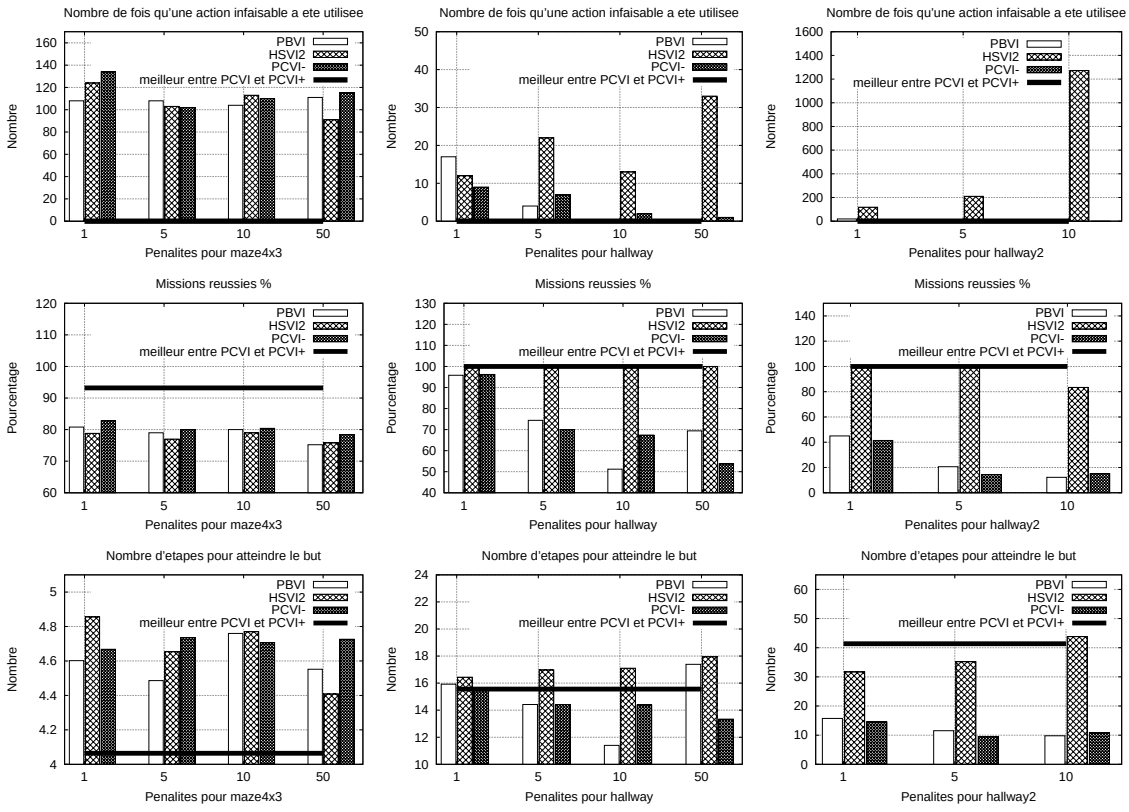


FIGURE 3 – Performance pour les problèmes de navigation ; PCVI- = états-support, PCVI = information additionnelle, PCVI+ = mélange (intersection) des deux autres.

quand les pénalité associées augmentent. Cependant, le seuil de la pénalité qui fait que l'agent ne choisit jamais une action infaisable n'est pas connu a priori.

Notre approche garantit qu'aucune action infaisable ne sera choisie indépendamment de la pénalité réglée – à la main – et associée avec la paire état-action irréalisable. En regardant le critère (2), nous pouvons vérifier que pour des pénalités importantes, le nombre de missions réussies diminue pour les politiques calculées avec PBVI, qui donne priorité à la non utilisation des actions infaisables, alors que HSVI2 relaxe les actions pour atteindre le but.

Notre approche atteint le but presque 100% du temps, même si ce critère n'est pas directement le but de notre modèle. En regardant le critère (3) pour le problème *maze4x3*, notre nouvelle étape de mise à jour de l'état de croyance réduit de manière très significative l'incertitude sur l'état de croyance de l'agent, et donc la moyenne d'étapes réalisées pour atteindre le but est aussi réduite. Pour les problèmes *hallway*, la moyenne du nombre d'étapes réalisées pour atteindre le but est légèrement plus importante que pour les autres algorithmes. Mais, nous remarquons que la moyenne est calculée en se basant sur le nombre de missions réussies, et ceci explique le fait que la politique issue de PBVI ait une bonne moyenne d'étapes réalisées pour atteindre le but et un mauvais pourcentage de réussite comparé avec les politiques issues de HSVI2 et PCVI.

5 Conclusion et travaux futurs

Dans cet article, nous proposons un nouveau modèle POMDP qui utilise des préconditions booléennes pour garantir que les politiques optimisées contiendront seulement des actions réalisables. Ceci requiert une adaptation du modèle de mise-à-jour de l'état de croyance ainsi qu'une adaptation de la mise à jour de la valeur par l'équation de Bellman. Nous avons, pour cette dernière, présenté un nouvel opérateur de *backup*, pour lequel nous démontrons que la propriété de contraction est respectée.

Les résultats de l'implémentation, avec une version modifiée de PBVI appelée PCVI, confirment que notre approche garantit de ne jamais appliquer des actions infaisables à l'exécution de la politique, indépendamment de la pénalité réglée "à la main" et associée aux paires état-action. De plus, notre approche montre des résultats compétitifs en termes d'élagages utiles dans des problèmes très contraints, de bons taux de réussite de mission et des récompenses cumulées.

Nous avons l'intention d'étendre notre approche et d'étudier d'autres opérateurs d'agrégation pour la fonction de valeur de l'état de croyance b comme fonction de la valeur des états de croyance successeurs $\tilde{b}$. Nous avons aussi l'intention d'étendre notre modèle à d'autres algorithmes de l'état de l'art tels que HSVI2 et SARSOP par exemple, qui auront besoin de l'adaptation du schéma de mise-à-jour de l'état de croyance, de la mise-à-jour de la valeur et des heuristiques pour les fonctions de préconditions booléennes.

Références

Cassandra A. (1998). *Exact and approximate algorithms for partially observable Markov decision processes*. PhD thesis, Brown University Providence, RI, USA.

Ghallab M., Nau D. & Traverso P. (2004). *Automated Planning : theory and practice*. Morgan Kaufmann.

Ingrand F., Lacroix S., Lemai-Chenevier S. & Py F. (2007). Decisional autonomy of planetary rovers. *Journal of Field Robotics*, **24**(7), 559–580.

Kurniawati H., Hsu D. & Lee W. (2008). SARSOP : Efficient point-based POMDP planning by approximating optimally reachable belief spaces. In *Proc. RSS*.

Pineau J., Gordon G. & Thrun S. (2003). Point-based value iteration : An anytime algorithm for POMDPs. In *Proc. of IJCAI*.

Smallwood R. & Sondik E. (1973). The optimal control of partially observable Markov processes over a finite horizon. *Operations Research*, p. 1071–1088.

Smith T. & Simmons R. G. (2005). Point-based POMDP algorithms : Improved analysis and implementation. In *Proc. UAI*.

Spaan M. & Vlassis N. (2004). A point-based POMDP algorithm for robot planning. In *ICRA*.

Teichteil-Königsbuch F., Kuter U. & Infantes G. (2010). Incremental plan aggregation for generating policies in MDPs. In *Proc. AAMAS*, p. 1231–1238.

Younes H. & Littman M. (2003). PPDDL1.0 : An extension to PDDL for expressing planning domains with probabilistic effects. In *Proc. of ICAPS*.