Trois Algorithmes pour Résoudre les N-POMDPs *

Yann Dujardin[1], Tom Dietterich[2], Iadine Chadès[1]

[1] CSIRO, Australia
{yann.dujardin,iadine.chades}@csiro.au
[2] EECS, Oregon State University, USA
tgd@oregonstate.edu

Mots-clés : *POMDP, planification dans l'incertain, environnement, solutions compactes*

1 Introduction

Les processus de décision Markoviens partiellement observables (POMDPs en anglais) sont un outil puissant pour modéliser les problèmes de décision séquentiels dans l'incertain. Dans le domaine des sciences informatiques appliquées à la protection de l'environnement (computational sustainability), les POMDPs sont utilisés de manière croissante pour aider à la résolution de problèmes importants en biologie de la conservation. Par exemple, des règles générales de surveillance et de gestion ont été trouvées en utilisant les POMDPs dans des cas de conservation d'espèces en danger critique tel que le tigre de Sumatra [1, 4], ou dans le cas d'espèces invasives telle que la plante parasite orobanche rameuse qui a des graines microscopiques à grande longevité [6]. Malgré ce regain d'intérêt, les solutions de POMDPs sont souvent trop complexes pour être analysées et communiquées efficacement aux décideurs, ce qui empêche le déploiment de telles solutions [9, 5]. Nous proposons ici de réduire le gap entre les solveurs POMDPs et les utilisateurs finaux en proposant des solutions approchées pouvant être représentées par un petit nombre d'alpha-vecteurs. L'algorithme α-min récemment publié [2] a été la première tentative pour résoudre les POMDPs avec une limite N sur la taille des solutions. Nous appelons ce problème N-POMDP. α-min est cependant un algorithme glouton et ne fournit pas de garanties de performances satisfaisantes. Ici nous proposons trois algorithmes tous basés sur la même approche de recherche de meilleure solution de POMDP de taille N. Dans le cas où une solution optimale est connue à l'avance, α-min-2-fast est capable de fournir de bonnes solutions de taille N très rapidement, et α-min-2-p fournit des solutions approchées de taille N avec garantie de performance. Enfin α-min-2-solve est un solveur heuristique de N-POMDP à part entière.

2 N-POMDPs

Un POMDP peut être vu comme une extension du modèle bien connu de processus de décision Markovien (MDP en anglais) au cas où l'état du système n'est pas directement observable mais où l'on recoit à chaque pas de temps de l'information sur l'état courant du systeme, permettant une mise à jour d'une "croyance" sur cette état, c'est-a-dire une distribution de probabilité [7]. En général, cette croyance est révisée à l'aide de règles bayésiennes. Nous appellerons S l'espace d'états et B (pour "beliefs") l'espace des croyances. Si on appelle A l'espace des actions, résoudre exactement un POMDP consiste à trouver une politique optimale Π qui à chaque état de croyance $b \in B$ fait correspondre une action $a \in A$, de manière à maximiser un objectif qui est la somme de récompenses sur un horizon de temps éventuellement infini. A l'instar des MDP, il existe une fonction de valeur V qui à chaque croyance initiale fait correspondre la valeur de l'objectif si on applique une politique optimale.

Les POMDPs peuvent être résolus en manipulant directement des α-vecteurs [8] : pour chaque POMDP, il existe un ensemble unique Γ de vecteurs de dimension $|S|$, appelés α-vecteurs, qui définissent entièrement $V : \forall b \in B, V(b) = \max_{\alpha \in \Gamma}(\alpha \cdot b)$. Par abus de langage, nous utiliserons donc également le terme "politique" (optimale) pour désigner Γ.

*Ce résume est une synthèse d'un article récemment accepté à la conférence AAAI-17 (mêmes auteurs)

Il devient alors aisé de définir formellement un N-POMDP (Definition 1).

Définition 1. *Un N-POMDP est un POMDP avec un paramètre additionnel N qui définit la taille maximum d'une politique admissible représentée par un ensemble d'α-vecteurs, et ce à chaque pas de temps (pour le cas de l'horizon infini nous considérons qu'il n'y a qu'un seul pas de temps).*

Résoudre un N-POMDP signifie trouver la meilleure politique possible de taille au plus N à chaque pas de temps. Dans le cas de l'horizon infini nous avons à résoudre le Problème 1 suivant.

Problème 1. *Calculer* $\max_{\Gamma \in \Theta,\ s.t. |\Gamma| \leq N} V_\Gamma(b)$ *et un ensemble* Γ *correspondant, où* V_Γ *est la fonction de valeur correspondant à la politique* Γ*, et* $b \in B$ *est une croyance initiale.*

Proposition 1. *Le problème 1 est NP-hard (la preuve de l'article original repose sur le fait que les POMDPs ne peuvent pas être résolus en un temps qui est polynomial en* $|\Gamma|$ *- preuve dans [3]).*

3 Les algorithmes α-min-2

Soit Γ une politique optimale d'un POMDP donne sans limitation de taille. Nous voulons résoudre le problème suivant.

Problème 2. *Calculer* $g^* = \min_{\tilde{\Gamma} \subseteq \Gamma, |\tilde{\Gamma}| \leq N} \max_{b \in B} [V(b) - \tilde{V}(b)]$ *et un* $\tilde{\Gamma}$ *correspondant.*

Soit s telle que : $s(\tilde{\alpha}, \alpha) = \max_{b \in B(\alpha)} (\alpha \cdot b - \tilde{\alpha} \cdot b), \alpha, \tilde{\alpha} \in \Gamma$, où $B(\alpha)$ est le sous-espace de B tel que α domine tous les autres α-vecteurs de Γ (i.e. meilleur sur toutes les composantes). Nous avons $\max_{b \in B} [V(b) - \tilde{V}(b)] \leq \max_{\alpha \in \Gamma} \min_{\tilde{\alpha} \in \tilde{\Gamma}} s(\tilde{\alpha}, \alpha)$. α-min-fast (Algorithme 1) consiste alors à résoudre le Problème 3.

Problème 3. *Calculer* $\min_{\tilde{\Gamma} \subseteq \Gamma, |\tilde{\Gamma}| \leq N} \max_{\alpha \in \Gamma} \min_{\tilde{\alpha} \in \tilde{\Gamma}} s(\tilde{\alpha}, \alpha)$ *et un* $\tilde{\Gamma}$ *correspondant.*

Résoudre le Problème 3 nous fournit alors une borne supérieure pour notre problème initial (Problème 2). C'est le but de l'Algorithme 1, dans lequel ϵ_{ub} (ligne 2) est une borne supérieure initiale de la solution ϵ du Problème 3 et $\tilde{\Gamma}$ (ligne 9) est la solution retournée par le programme linéaire LP_{FAST} suivant : $\min \sum_{\alpha \in \Gamma} x_\alpha \ s.t. \ \forall \alpha' \in \Gamma, \ \sum_{\alpha \in \Gamma} s(\alpha, \alpha') x_\alpha \geq 1, \sum_{\alpha \in \Gamma} x_\alpha \leq N, x_\alpha \in \{0, 1\}$. La politique est construite en respectant la règle suivante : $\alpha \in \tilde{\Gamma} \iff x^*_\alpha = 1$.

Proposition 2. *L'Algorithme 1 résout le Problème 3 avec une précision donnée* p *en temps* $O(\log(\frac{\epsilon_{ub}}{p}) P(|\Gamma|, log(N)) 2^{|\Gamma|})$*, où* P *est un polynôme (voir la preuve dans l'article original).*

α-min-2-p (Algorithme 2) maintient quant à lui sur g^* à la fois une borne inférieure $g(\Delta) = \min_{\tilde{\Gamma} \subseteq \Gamma, |\tilde{\Gamma}| \leq N} \max_{\beta \in \Delta} \min_{\tilde{\alpha} \in \tilde{\Gamma}} (\alpha \cdot b - V(b))$ (avec une precision donnée p) et une borne supérieure $g_{ub} = \max_{b \in B} (V(b) - \tilde{V}(b))$. Δ est un sous-sespace de B qui est itérativement améliore en ajoutant progressivement les croyances telles que g_{ub} et $g(\Delta)$ soient de plus en plus proches. Le processus se termine lorsque leur différence atteint $\frac{p}{2}$ ce qui permet de calculer une garantie de performance en p.

Algorithm 1 α-min-2-fast

1: **procedure** FAST($\Gamma, p, N \geq 1$)
2: $\epsilon^+ = \epsilon_{ub}, \epsilon^- = 0, \delta = \epsilon^+ - \epsilon^-$
3: **while** $\delta > p$ **do**
4: $\epsilon = \frac{\epsilon^+ + \epsilon^-}{2}$
5: **for** $\alpha, \alpha' \in \Gamma$ **do**
6: **if** $s_{\alpha,\alpha'} \leq \epsilon$ **then** $c_{\alpha,\alpha'} = 1$
7: **else** $c_{\alpha,\alpha'} = 0$
8: $C = \{c_{\alpha,\alpha'} \mid \alpha, \alpha' \in \Gamma\}$
9: $\tilde{\Gamma} \leftarrow LP_{FAST}(C, N)$
10: **if** $LP_{FAST}(C, N)$ has no solution **then** $\epsilon^+ = \epsilon$
11: **else** $\epsilon^- = \epsilon$
12: $\epsilon^+ = \epsilon_{ub}, \epsilon^- = 0, \delta = \epsilon^+ - \epsilon^-$
return $\tilde{\Gamma}, \epsilon$

Algorithm 2 α-min-2-p

1: **procedure** PRECISE(Γ, p, N)
2: Let Δ be the set $(1, 0, \cdots, 0), (0, 1, \cdots, 0), \cdots, (0, \cdots, 0, 1)$ (extreme beliefs of B)
3: $\delta \leftarrow \inf, g_{ub} \leftarrow \inf$
4: **while** $\delta > \frac{p}{2}$ **do**
5: $(\tilde{\Gamma}, g_\Delta) \leftarrow FAST_\beta(\Gamma \times \Delta, \frac{p}{2}, N)$
6: $b^* \leftarrow \arg\max_{b \in B} (V(b) - \tilde{V}(b))$
7: $g_{ub} \leftarrow \min(g_{ub}, V(b^*) - \tilde{V}(b^*))$
8: $\delta \leftarrow g_{ub} - g_\Delta$
9: $\Delta \leftarrow \Delta \cup \{b^*\}$
return $\tilde{\Gamma}, g_{ub}$

Proposition 3. *L'Algorithme 2 résout le Problème 2 avec une précision donnée p en temps $O(|\Gamma|^N \log(\frac{\epsilon_{ub}}{p}) P(|\Gamma|, log(N)) 2^{|\Gamma|})$, où P est un polynôme (voir la preuve dans l'article original).*

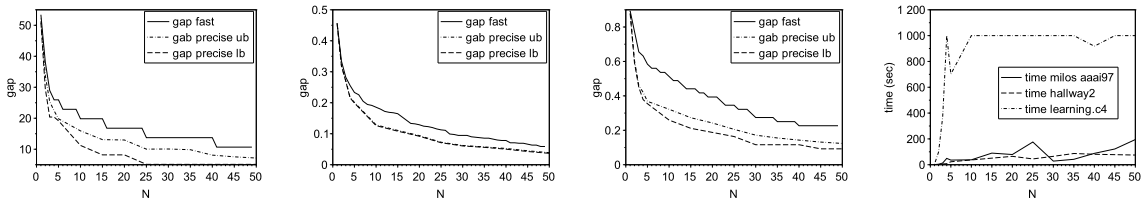
Enfin, α-min-2-solve est une version heuristique de α-min-2-p où la ligne 6 de l'Algorithme 2 est remplacée par une génération aléatoire de croyances, ce qui permet à α-min-2-solve d'être utilisé en tant que solveur (sans recours à une politique initiale). La raison principale est que la ligne 6 est maintenant facile à réaliser même dans le cas où V n'est pas connu à l'avance. α-min-2-solve a la même complexité au pire cas que α-min-2-p. Plus de détails sont disponible dans l'article original.

4 Expérimentations

Nous comparons dans cette section α-min-2-fast à α-min-2-p, qui ont tous deux besoin de politique initiale, et α-min-2-solve à α-min, qui sont tous deux des solveurs de N-POMDP.

Pour comparer α-min-2-fast et α-min-2-p, nous avons conduit des expérimentations sur trois benchmarks [1], avec une politique initiale calculée par le solveur SARSOP. Pour chaque problème, nous notons par $(|S|, |A|, |O|, |\Gamma|, \underline{V}(b_0)))$ ses caractéristiques, où $\underline{V}(b_0)$ est une borne inférieure très proche de la fonction de valeur $V(b_0)$ du POMDP en un croyance initiale b_0, et Γ est la solution correspondante. Tous deux sont calculés par SARSOP. Nous avons les benchmarks milos-aaai997 $\equiv$ (20, 6, 8, 184, 574.8), hallway2 $\equiv$ (92, 5, 17, 139, 0.25) et learning.c4 $\equiv$ (48, 16, 3, 332, 3.3).

Les Figures (1a, 1b et 1c) montrent les profiles encourageants des gaps et bornes en fonction de N. Pour α-min-2-fast seule une borne supérieure de g^* est disponible, appelée "gap fast". Pour α-min-2-p la borne supérieure g_{ub} et la borne inférieure $g(\Delta) - \frac{p}{2}$ sont fournis et sont appeles respectivement "gap precise ub" et "gap precise lb". La Figure (1d) fournit les temps de calcul pour α-min-2-p. Les temps ne sont pas indiqués pour α-min-2-fast car négligeables (<1 seconde).



(a) Gaps milos-aaai97 (b) Gaps hallway2 (gap_{ub} et gap_{lb} se confondent presque parfaitement) (c) Gaps learning.c4 (d) Temps de calcul (α-min-2-p)

FIG. 1

Nous avons comparé α-min-2-solve à α-min sur des benchmarks de [2] (horizon fini de $T = 10$). La Table (1) montre que α-min-2-solve est toujours plus rapide que α-min alors qu'il fournit de meilleures solutions (LB). dujardin-ijcai15 réfère au problème : "the four populations Sumatran tigers non-stationary problem" [2], disponible sur https ://sites.google.com/site/ijcaialphamin/home.

Problème $(\|S\|, \|A\|, \|O\|)$	**Algo.**	N	**LB**	**Temps(s)**	**Problème**	**Algo.**	N	**LB**	**Temps(s)**
aloha.10	sarsop	190	64.87	1000	cheng.D4-5	sarsop	15	77.29	1000
(30,9,3)	α-min	30	62.66	$\leq$1000	(4,4,4)	α-min	4	77.85	$\leq$1000
	α-min-2	30	**63.51**	$<$ **1**		α-min-2	4	**77.90**	$<$ **1**
learning.c3	sarsop	11433	1.36	1000	milos-aaai97	sarsop	122	41.48	1000
(24,12,3)	α-min	24	1.96	$\leq$1000	(20,6,8)	α-min	20	50.31	$\leq$1000
	α-min-2	24	**2.09**	$<$ **1**		α-min-2	20	**54.76**	$<$ **1**
dujardin-ijcai15	α-min	7	207.23	37.8					
(16,13,16)	α-min-2	7	**208.45**	$<$ **1**					

TAB. 1 – Comparison de α-min-2-solve et α-min. LB (lower bound) est la valeur fournie pour approcher $V(b_0)$.

1. disponibles sur http ://www.pomdp.org/examples/

Enfin, α-min-2-fast and α-min-2-p sont capables de résoudre des problèmes plus grands, tel que TagAvoid $\equiv$ $(870,\ 5,\ 30,\ 615,\ -6.76)$. α-min-2-fast permet d'atteindre $LB \geq 0.98\ \underline{V}(b_0)$ très rapidement (moins de 3 secondes) mais sans garantie de performance, alors que α-min-2-p le résout en 3389 secondes mais avec une garantie (par example avec une précision de 0.7 pour $N = 10$).

5 Discussion

Les trois algorithmes présentés dans ce résumé résolvent de manière approchée les N-POMDPs avec différents niveau d'approximation et d'hypothèses. α-min-2-fast minimise une borne supérieure du gap entre une politique initiale, fournie par un solveur extérieur, et une politique utilisant seulement N α-vecteurs. α-min-2-p utilise à la fois une borne supérieure et inférieure, ce qui lui permet de fournir des solutions avec une précision choisie. Cependant, il est clairement plus lent que α-min-2-fast. Enfin, α-min-2-solve a été écrit pour pouvoir résoudre des N-POMDPs sans avoir besoin de politique initiale Γ du POMDP. Cependant, α-min-2-solve est une heuristique, et ne fonctionne qu'en horizon fini (la convergence n'est pas forcément assurée en horizon infini).

Pour chacun des trois algorithmes, le temps de calcul n'est pas très sensible à N, alors que la complexité au pire cas indique une dépendance exponentielle en N. En pratique, c'est plutôt la taille de la politique initiale Γ qui semble jouer un rôle prédominant dans le temps de calcul, notamment de α-min-2-p. En effet, α-min-2-p doit résoudre Γ programmes linéaires à chaque itération. Des travaux futurs auront pour but de réduire le temps de calcul pratique.

Références

[1] I. Chadès, E. McDonald-Madden, M. A. McCarthy, B. Wintle, M. Linkie, and H. P. Possingham. When to stop managing or surveying cryptic threatened species. *Proceedings of the National Academy of Sciences*, 105(37) :13936–13940, 2008.

[2] Y. Dujardin, T. Dietterich, and I. Chadès. α-min : A compact approximate solver for finite-horizon POMDPs. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI 15)*, pages 2582–2588, 2015.

[3] M. L. Littman, A. R. Cassandra, and L. P. Kaelbling. Efficient dynamic-programming updates in partially observable Markov decision processes. Technical report, 1995.

[4] E. McDonald-Madden, I. Chadès, M. A McCarthy, M. Linkie, and H. P. Possingham. Allocating conservation resources between areas where persistence of a species is uncertain. *Ecological Applications*, 21(3) :844–858, 2011.

[5] S. Nicol and I. Chadès. Which states matter ? An application of an intelligent discretization method to solve a continuous POMDP in conservation biology. *PLoS ONE*, 7(2) :e28993, 02 2012.

[6] T. J Regan, I. Chadès, and H. P. Possingham. Optimally managing under imperfect detection : a method for plant invasions. *Journal of Applied Ecology*, 48(1) :76–85, 2011.

[7] O. Sigaud and O. Buffet. *Markov decision processes in artificial intelligence*. John Wiley & Sons, 2013.

[8] R. D. Smallwood and E. J. Sondik. The optimal control of partially observable Markov processes over a finite horizon . *Operations Research*, 21(5) :1071–1088, 1973.

[9] V. J.D. Tulloch, A. I.T. Tulloch, P. Visconti, B. S. Halpern, J. E.M. Watson, M. C. Evans, N. A. Auerbach, M. Barnes, M. Beger, I. Chadès, et al. Why do we map threats ? Linking threat mapping with actions to make better conservation decisions. *Frontiers in Ecology and the Environment*, 13 :91–99, 2015.