Programmation dynamique à mémoire bornée avec distribution sur les croyances pour les Dec-POMDPs

Gabriel Corona, François Charpillet

Programmation dynamique à mémoire bornée avec distribution sur les croyances pour les Dec-POMDPs

Gabriel Corona, François Charpillet

INRIA, LORIA
Campus Scientifique, BP 239
54506 Vandœuvre-lès-Nancy Cedex
prénom.nom@loria.fr

Résumé : Nous proposons une approche heuristique pour calculer une politique approchée d'un Dec-POMDP. Il s'agit d'une approche par programmation dynamique à base de points dans la lignée des algorithmes PBDP (Szer & Charpillet, 2006), MBDP (Seuken & Zilberstein, 2007b) et IMBDP (Seuken & Zilberstein, 2007a) : Elle formule le choix des politiques retenues à chaque étape de la construction comme un problème d'optimisation. Le critère de ce problème repose sur une estimation de la distribution de probabilité *a priori* des croyances atteignables pour un horizon donné : Il s'agit de maximiser l'espérance des récompenses cumulées pour l'horizon considéré étant donné cette distribution. L'estimation de cette espérance peut se faire par échantillonnage des croyances en simulant une politique heuristique.

Mots-clés : Dec-POMDP, planification, multi-agent, programmation dynamique, optimisation combinatoire, Monte-Carlo, Importance Sampling.

1 Introduction

Le problème du contrôle d'une équipe d'agents coopératifs interagissant avec un environnement stochastique et partiellement observable possède un large champs applicatif : routage dans un réseau, coordination de robots, surveillance de zone, *smart surface* ...

Le cadre des Processus décisionnel de Markov décentralisés partiellement observables, Dec-POMDP, permet d'aborder cette problématique de manière rigoureuse. Dans ce papier, nous nous intéressons plus particulièrement à la planification, c'est-à-dire au calcul d'une politique (idéalement optimale) pour chacun des agents en connaissant les lois du système et la fonction de récompense. Cependant, le calcul d'une politique optimale se heurte à des problèmes théoriques car il est NEXP-complet en horizon fini (Bernstein *et al.*, 2000) ce qui rend sa résolution exacte dans le cas général irréalisable sauf pour des problèmes de petite taille et de faible horizon.

Au moins trois grandes classes d'approches existent pour résoudre un Dec-POMDP : Les approches ascendantes par programmation dynamique (Hansen *et al.*, 2004), les approches descendantes (Szer & Charpillet, 2005) et les approches par programmation mathématique (Aras *et al.*, 2007). Nous nous concentrerons dans la suite sur le premier type d'approche.

Le nombre de politiques partielles à considérer dans une approche par programmation dynamique exacte croît dans le cas général exponentiellement avec l'horizon de planification et avec le nombre d'observations ce qui limite en pratique son application à des horizons et des nombres d'observation petits. Pour pallier ces problèmes, des algorithmes approchés ont été proposés : MBDP (Seuken & Zilberstein, 2007b), IMBDP (Seuken & Zilberstein, 2007a), MBDP-OC (Carlin & Zilberstein, 2008) ... Ils utilisent une heuristique pour ne retenir à chaque étape de la programmation dynamique qu'un petit nombre de plans partiels. En bornant l'espace mémoire nécessaire au calcul, ils permettent de traiter des problèmes significativement plus grands que les algorithmes antérieurs en terme d'horizon et, en évitant l'étape de mise à jour exhaustive, de nombres d'observations. Cependant, les solutions fournies par ces algorithmes ne présentent pas de garantie théorique de performance.

Dans le domaine des MDPs, l'algorithme PSDP (Bagnell *et al.*, 2003) utilise une distribution heuristique à priori sur les états pour guider la recherche d'une politique au sein d'une classe restreinte de politiques : Si

aucune borne d'erreur pratique n'est fournie, un argument théorique est avancé sur la qualité de la solution obtenue quand la distribution heuristique est proche de celle obtenue en effectuant une « bonne » politique.

L'approche proposée concilie ces deux approches : Nous proposons de généraliser l'approche de PSDP aux cas POMDP et Dec-POMDP en considérant la distribution à priori sur les croyances ce qui peut se ramener à un algorithme à base de points : Le choix des plans partiels à retenir devient un problème d'optimisation combinatoire (maximisation de la récompense espérée).

De cette manière, nous envisageons de traiter des problèmes de grande taille et d'essayer d'adapter la garantie théorique de Bagnell *et al.* (2003) pour les Dec-POMDPs.

2 Modèle Dec-POMDP

2.1 Modèle

Un Dec-POMDP (en horizon fini) peut se définir sous forme d'un multiplet $\langle I, S, A, \mathcal{P}, \mathcal{R}, O, \mathcal{O}, T, b_0 \rangle$, où

- $I = \{1, \ldots, n\}$ est l'ensemble fini des n agents (notés i) ;
- S est un ensemble fini d'états notés s ;
- A est l'ensemble des actions jointes notées $a = (a_1, \ldots a_n)$ où les $a_i \in A_i$ sont les actions locales de l'agent i ;
- $\mathcal{P} : S \times A \to \Delta S$ définit la probabilité de transition, $\Pr(s'|s, a)$;
- $R : S \times A \to \mathbb{R}$ est une fonction de récompense qui définit l'espérance de la récompense, $R(s, a)$, quand on effectue l'action a dans l'état s ;
- O est l'ensemble des observations jointes notées $o = (o_1, \ldots, o_n)$ où les $o_i \in O_i$ sont les observations locales de l'agent i ;
- $\mathcal{O} : S \times A \times S \to \Delta O$ définit la probabilité d'observations, $\Pr(o|s, a, s')$;
- T est l'horizon du problème ;
- $b_0 \in \Delta S$ est la distribution initiale sur les états.

Résoudre un Dec-POMDP pour un horizon fini T étant donnée une distribution d'états initiale b_0 revient à trouver un ensemble de n politiques locales q_i (une pour chaque agent) de manière à maximiser l'espérance de la somme des récompenses du système. Si on note, q la sous-politique jointe formée par les q_i, le critère est :

$$\max \mathrm{E}\left[\sum_{t=0}^{T-1} R(s_t, a_t)|b_0, q\right].$$

2.2 Arbres de politique

Rappelons qu'en horizon fini les politiques locales peuvent être représentées sous forme d'arbres de décision (figure 1) dont les nœuds sont des actions et les branches des observations. Nous notons q_i un tel arbre de décision et Q_i un ensemble d'arbres pour l'agent i. On note $Q = \prod_i Q_i$ un ensemble de politiques jointes.

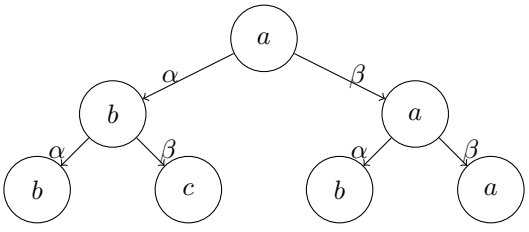

FIG. 1 – Exemple de politique pour un agent i, $T = 3$, $O_i = \{\alpha, \beta\}$, $A_i = \{a, b, c\}$

Une politique jointe q est n-uplet de politiques jointes (une pour chaque agent) : $q = (q_1, \ldots, q_n)$. L'évaluation d'une politique jointe se fait par programmation dynamique :

$$\forall s, V_q(s) = R(s, a(q)) + \sum_{o \in O, s' \in S} P(s', o|s, q) V_{q(o)}(s') \quad (1)$$

où $a(q)$ représente l'action racine de q, o une observation jointe et $q(o)$ la politique jointe $q' = (q'_0, \ldots, q'_n)$ composée des sous-arbres q'_i pour les observations o_i. La fonction de valeur V_q d'une politique jointe q peut être représentée sous la forme d'un $|S|$-vecteur α_q : $\alpha_{qs} = V_q(s)$.

De plus, une distribution de probabilités sur les états est un $|S|$-vecteur $b \in \Delta S$: La valeur d'une politique q étant donnée une telle distribution prend la forme :

$$V_q(b) = \sum_{s \in S} b(s) V_q(s) = b \cdot \alpha_q.$$

2.3 POMDP

Un POMDP peut être vu comme un cas particulier de Dec-POMDP avec un seul agent : $n = 1$, $I = \{1\}$, $O = O_1$, $A = A_1$. Dans un POMDP, cette distribution b peut en particulier représenter la *croyance* de l'agent sur l'état du système étant donné la séquence des observations passées.

Pour un POMDP, un ensemble Q d'arbres q définit une politique : Pour toute croyance b, l'arbre q qui maximise $V_q(b)$ est utilisé. La fonction de valeur de cette politique Q est linéaire par morceaux et convexe :

$$\forall b, V_Q(b) = \max_{q \in Q} V_q(b).$$

Étant donné un Dec-POMDP, on peut introduire le POMDP sous-jacent : Il s'agit du POMDP obtenu en remplaçant les n agents pas un agent unique qui observe les observations jointes et choisit les actions jointes.

2.4 MDP

Un MDP peut être vu comme un cas particulier de POMDP où l'agent observe directement les états : $O = S$ et $\Pr(o = s' | s, a, s') = 1$.

Étant donné un Dec-POMDP, on peut définir le MDP sous-jacent comme étant le MDP qui a les mêmes lois de transition et de récompense.

3 Contexte

3.1 Approches à base de points

Les algorithmes à base de points MBDP (Seuken & Zilberstein, 2007b) et IMBDP (Seuken & Zilberstein, 2007a) ont récemment été proposés pour résoudre des Dec-POMDPs avec de grands horizons et (pour IMBDP) avec un grand nombre d'observations : La complexité est linéaire par rapport à l'horizon et exponentielle (pour MBDP) ou polynomiale (pour IMBDP) par rapport au nombre d'observations.

Ces deux algorithmes construisent les politiques par programmation dynamique de manière itérative. La première itération recherche pour chaque agent i des politiques d'horizon 1, puis à chaque itération, l'algorithme considère un ensemble Q_i^t de politiques d'horizon $T - t$ représentées sous forme d'arbres.

Les ensembles $\bar{Q}_i^t$ des politiques locales d'horizon $T - t + 1$ sont générés de manière exhaustive (pour le cas de MBDP[1]) en construisant de nouveaux arbres dont la racine est une des actions possibles et chaque branche un des arbres de Q_i^{t+1} retenus à l'itération précédente : $\bar{Q}_i^t = A_i \times (Q_i^{t+1})^{O_i}$. Après cette étape, suit une phase d'élagage heuristique : Pour chaque agent, seuls $\mathrm{maxTrees}$ arbres sont conservés pour former l'ensemble Q_i^t.

Pour cela, $\mathrm{maxTrees}$ distributions $b \in \Delta S$ *à priori* sur les états sont générées en suivant divers heuristiques : Les politiques jointes $q^t = (q_1^t, \ldots, q_n^t)$ sont évaluées pour chaque b et la meilleure est retenue. Au final, $\mathrm{maxTrees}$ politique jointes q^t sont retenues et donc $\mathrm{maxTrees}$ politiques locales q_i^t pour chaque agent i qui forment les ensembles Q_i^t.

[1] IMBDP remplace la mise à jour exhaustive de complexité exponentielle par rapport au nombre d'observations par une mise à jour partielle qui ne génère qu'un sous ensemble des arbres possibles.

3.2 PSDP

L'algorithme PSDP (Bagnell *et al.*, 2003) cherche une politique déterministe de MDP dans un sous-espace restreint de politiques donné : La politique cherchée $\pi = \left(\pi^0, \ldots, \pi^{T-1}\right) \in \Pi^T$ est est une politique non-stationnaire composée d'une série de politiques stationnaires $\pi^t \in \Pi$. Π et Π^T sont les espaces de recherche (respctivement des π^t et de π) et forment un sous-espace de l'espace des politiques possibles.

PSDP résout ce problème par programmation dynamique bottom-up : Étant donné π^{t+1} jusqu'à $\pi^{(T-1)}$, on cherche $\pi^t \in \Pi$ et donc $\left(\pi^t, \ldots \pi^{T-1}\right) \in \Pi^{T-t}$. On peut introduire un pré-ordre non total entre les éléments de Π^{T-t},

$$\psi = \left(\pi^t, \ldots \pi^{T-1}\right) \prec \psi' = \left(\pi'^t, \ldots \pi^{T-1}\right) \Leftrightarrow \forall s, V_\psi(s) \leq V_{\psi'}(s).$$

En d'autres termes, on ne peut pas choisir une politique π^t plutôt qu'une autre car, dans le cas général, aucune politique ne domine toutes les autres.

Pour résoudre ce problème, PSDP introduit une connaissance supplémentaires. L'introduction d'une distribution heuristique (à priori), $\mu_t \in \Delta S$, sur les états s^t permet de rendre le pré-ordre total :

$$\psi \prec \psi' \Leftrightarrow \mathop{\mathrm{E}}_{s \sim \mu^t}\left[V_\psi(s)\right] \leq \mathop{\mathrm{E}}_{s \sim \mu^t}\left[V_{\psi'}(s)\right].$$

En utilisant cette connaissance heuristique, on peut donc choisir π_t :

$$\pi_t = \arg\max_{\pi' \in \Pi} \mathop{\mathrm{E}}_{s \sim \mu^t}\left[V_{\pi', \pi^{t+1}, \ldots \pi^{T-1}}(s)\right].$$

4 Approche proposée

4.1 Présentation générale

Un POMDP peut être vu comme un MDP ayant pour espace d'état S' l'espace continu des croyances (Kaelbling *et al.*, 1995) : $S' = \Delta S$. De plus, la planification dans le cadre Dec-POMDP peut être vue comme la recherche d'une politique de POMDP dans le sous-espace des politiques contraintes par le décentralisation de décisions. Il semble donc naturel de chercher à adapter la méthode de PSDP pour les cas POMDP et Dec-POMDP en introduisant des distribution heuristiques μ^t *sur les croyances* b^t (du POMDP ou du POMDP sous-jacent au Dec-POMDP) : Une politique donnée génère une distribution de probabilité sur les séquences d'observations possibles et donc sur les croyances aux différents instants.

On voudrait donc considérer une méthode à base de points qui s'intéresse aux distributions à posteriori sur les états, les croyances, plutôt qu'aux distributions à priori. Les points b^t sont échantillonnés selon μ^t et l'ensemble des points peut être considéré comme une approximation par échantillonnage stochastique de μ^t.

Plutôt que de sélectionner chaque arbre de politique à retenir en ne se basant que sur un seul échantillon, nous proposons donc de choisir les ensembles Q_i^t de $\mathrm{maxTrees}$ arbres pour chaque agent i en utilisant le plus grand nombre possible d'échantillons c'est-à-dire la meilleure approximation possible de μ^t : La vraie distribution sur les croyances ne peut être connue avant d'avoir terminé la planification car elle dépends de la politique utilisée et on peut donc utiliser une « bonne » politique heuristique pour générer les distributions heuristiques μ^t.

Le choix de l'ensemble Q_i^t de $\mathrm{maxTrees}$ arbres par agent parmi les $|A_i|\mathrm{maxTrees}^{|O_i|}$ arbres possibles générés par mise à jour exhaustive devient un problème d'optimisation combinatoire pour maximiser la valeur. Si on note $\mathbb{Q}_i^t$ les ensembles des combinaisons possibles d'éléments de $\bar{Q}_i^t$ de taille inférieure ou égale à $\mathrm{maxTrees}$,

$$\forall i, \mathbb{Q}_i^t = \left\{Q_i^t / Q_i^t \subset \bar{Q}_i^t, |Q_i^t| \leq \mathrm{maxTrees}\right\},$$

alors $\mathbb{Q}^t = \prod_{i=1}^n \mathbb{Q}_i^t$ est l'espace de recherche et le critère devient :

$$\max_{Q^t \in \mathbb{Q}^t} \mathop{\mathrm{E}}_{b^t \sim \mu^t}\left[V_{Q^t}(b^{t-1})\right] \text{ avec } V_{Q^t}(b) = \max_{\forall i, q_i^t \in Q_i^t} V_{q_i^t}(b).$$

4.2 Idée intuitive

Pour simplifier la présentation, restreignons-nous au cas à un agent. Une première approche pour la sélection des arbres consiste à introduire une mesure de la perte générée par la réduction de l'ensemble des arbres obtenus par mise à jour exhaustive $\bar{Q}^t = A \times (Q^{t+1})^O$ à Q^t. Intuitivement (voir figure 2), il s'agit de l'aire entre les courbes de leurs fonctions de valeur $\bar{V}^t$ et V^t :

$$\delta(\bar{V}^t, V^t) = \int \left(\bar{V}^t(b) - V^t(b)\right)\ db.$$

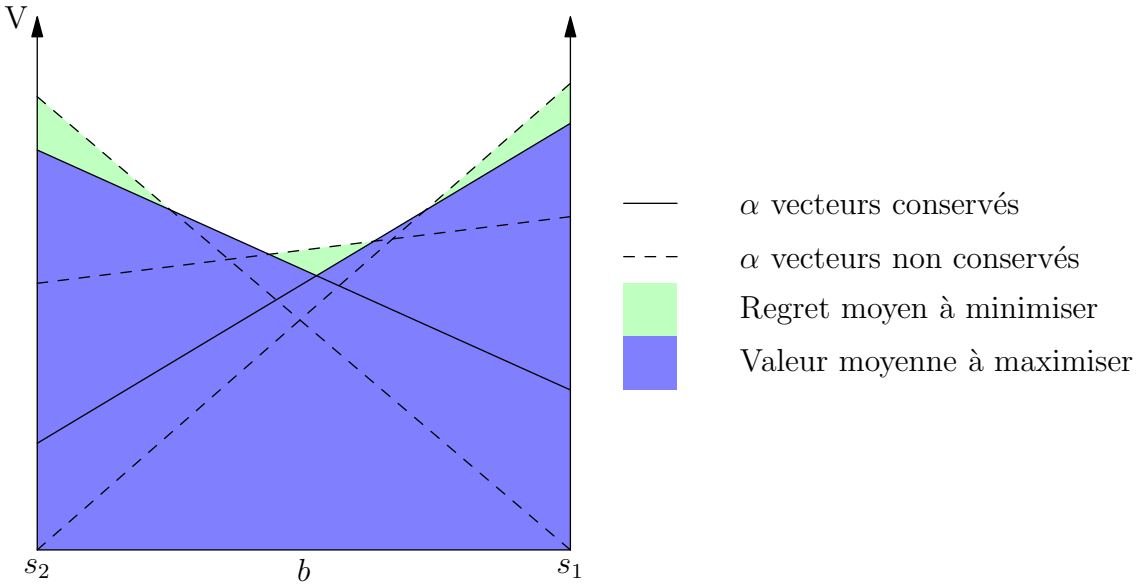

FIG. 2 – Intégration sur les croyances

Le choix des maxTrees arbres peut dans un premier temps être formulé comme un problème de minimisation de cette perte ou de manière équivalente à la maximisation de l'intégrale $I^t(V^t)$ de V^t :

$$\max_{V^t} I^t\left[V^t(b)\right] = \max_{V} \int V^t(b)\ db.$$

Cependant, dans la pratique, si on fixe la croyance b^0 initiale[2], seul un nombre discret de croyances est accessible pour un POMDP (ou un Dec-POMDP) fini : Tous les états de croyance intégrés dans le critère ne sont donc pas intéressants.

Pour généraliser cette idée, toutes les croyances n'ont pas la même importance car certaines se produisent avec une plus forte probabilité que d'autres. Ceci conduit à introduire une distribution de probabilité μ^t sur les différentes croyances dans le critère précédent avec la fonction d'objectif :

$$I^t(V^t) = \int V^t(b^t)\mu^t(b^t)\ db^t = \underset{b^t \sim \mu^t}{\mathrm{E}}\left[V^t(b^t)\right].$$

4.3 Distribution heuristique

Le choix de la distribution heuristique μ^t utilisée est une des variables de ce principe général et nous nous intéressons ici plus particulièrement à la distribution induite par une politique heuristique et une croyance initiale[3]. Cette distribution est une distribution finie et l'intégration peut théoriquement être effectuée de manière exacte,

$$I^t(V^t) = \sum_{b^t \in \mathcal{B}^t(\tilde{q}, b^0)} Pr(b^t|b^0, q)V^t(b^t),$$

[2] où plus généralement une distribution discrète de b^0

[3] D'autres possibilités pourraient être considérée. En particulier, le choix de la distribution pourrait permettre d'introduire une certaine forme de connaissance experte à priori sur le problème à résoudre).

où $\mathcal{B}^t(\tilde{q}, b^0)$ est l'ensemble des croyances du POMDP sous-jacent atteignables à l'instant t en suivant la politique heuristique $\tilde{q}$ à partir de b_0. Cet ensemble est cependant de taille exponentielle par rapport à l'horizon ce qui limite l'utilisation de l'évaluation exacte.

Le critère est donc approché par échantillonnage sur la distribution heuristique sur les croyances (approche Monte-Carlo),

$$I^t(V^t) \approx \hat{I}^t(V^t) = \frac{1}{N} \sum_{k=1}^{N} V^t(b_k^t),$$

où les b_k sont des échantillons indépendants de la distribution μ^t. Chaque échantillon peut être obtenu en simulant la politique heuristique jusqu'à l'instant t.

Des méthodes d'échantillonnage plus sophistiquées, comme Importance Sampling (Doucet *et al.*, 2001), pourraient être utilisées afin de réduire l'erreur de l'estimation (la variance) en générant plus d'échantillons sur les trajectoires qui semblent les plus intéressantes mais cette approche n'est pas développée ici.

4.4 Algorithme

L'algorithme 1 utilise l'échantillonnage de Monte-Carlo de la distribution heuristique pour estimer la valeur des politiques. Le problème d'optimisation combinatoire est résolu de manière approchée.

Algorithme 1 : PS-MBDP

Données : Dec-POMDP P
Données : Politique heuristique π
Données : Nombre de trajectoires N
Données : Nombre d'arbres par agent $\mathrm{maxTrees}$
Résultat : $\forall i \in I, Q_i^0$, ensemble d'arbres de politiques (et les vecteurs α associés)
début
  Simuler N trajectoires et retenir les croyances b_k^t
  $\forall i \in I, Q_i^T = \{\psi\}$ (arbre vide)
  pour *t=T-1* **à** *0* **faire**
    Chercher $Q_i^t, \dots Q_j^t$ qui maximisent $\hat{I}^t(V_{Q^t})$
    $\forall q \in \prod_i Q_i^t$, calculer α_q
  fin
fin

4.5 Éviter la mise à jour exhaustive

La mise à jour exhaustive, c'est-à-dire la construction de l'ensemble $\bar{Q}_i^t = A_i \times \left(Q_i^{t+1}\right)^{O_i}$, a une complexité exponentielle par rapport au nombre d'observations ce qui limite son utilisation aux problème avec un faible nombre d'observations. Des techniques de mise à jour partielles sont utilisées par IMBDP (Seuken & Zilberstein, 2007a) et MBDP-OC (Carlin & Zilberstein, 2008) pour traiter des problèmes avec un nombre conséquent d'observations.

Une piste intéressant consiste à chercher directement dans l'ensemble $\mathbb{Q}^{t-1} = \prod_i \mathbb{Q}_i^{t-1}$ en évitant l'étape de mise à jour qui consiste à générer explicitement les ensembles $\bar{Q}_i^t$. Éviter la mise à jour exhaustive se ferait au détriment de la qualité de l'étape d'optimisation mais un plus grand nombre d'arbres pourrait être conservé.

5 Conclusion

Nous avons proposé d'introduire un critère d'optimisation pour sélectionner les arbres de politiques dans une approche par programmation dynamique approchée à mémoire bornée : la maximisation de la récompense espérée étant donné la distribution sur les croyances. Cependant, la vraie distribution sur les croyances ne peut pas être déterminée avant d'avoir calculé la politique globale. Une distribution heuristique est donc utilisée. Si nous proposons ici d'utiliser une politique heuristique pour générer cette distribution, d'autres approches pourraient être envisagées.

En utilisant l'approche de PSDP, nous essayons de voir si la garantie théorique fournie par celui-ci peut être adaptée à la programmation dynamique à base de points à mémoire bornée pour les Dec-POMDPs.

Références

ARAS R., DUTECH A. & CHARPILLET F. (2007). Mixed integer linear programming for exact finite-horizon planning in decentralized POMDPs. In *Proceedings of the Thirteenth International Conference on Automated Planning and Scheduling (ICAPS 2007)*.

BAGNELL J. D., KAKADE S., NG A. & SCHNEIDER J. (2003). Policy search by dynamic programming. In *Neural Information Processing Systems*, volume 16 : MIT Press.

BERNSTEIN D. S., ZILBERSTEIN S. & IMMERMAN N. (2000). The complexity of decentralized control of markov decision processes. In *Mathematics of Operations Research*, p. 2002.

CARLIN A. & ZILBERSTEIN S. (2008). Value-based observation compression for dec-pomdps. In *AAMAS '08 : Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems*, p. 501–508, Richland, SC : International Foundation for Autonomous Agents and Multiagent Systems.

A. DOUCET, N. DE FREITAS & N. GORDON, Eds. (2001). *Sequential Monte Carlo methods in practice*.

HANSEN E., BERNSTEIN D. & ZILBERSTEIN S. (2004). Dynamic programming for partially observable stochastic games. In *In Proceedings of the Nineteenth National Conference on Artificial Intelligence*, p. 709–715.

KAELBLING L. P., LITTMAN M. L. & CASSANDRA A. R. (1995). Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, **101**, 99–134.

SEUKEN S. & ZILBERSTEIN S. (2007a). Improved Memory-Bounded Dynamic Programming for Decentralized POMDPs. Vancouver, BC, Canada.

SEUKEN S. & ZILBERSTEIN S. (2007b). Memory-Bounded Dynamic Programming for DEC-POMDPs.

SZER D. & CHARPILLET F. (2005). Point-based dynamic programming for DEC-POMDPs. In *Proceedings of the Twenty-First Conference on Uncertainty in Artificial Intelligence*.

SZER D. & CHARPILLET F. (2006). Programmation dynamique à base de points pour la résolution des dec-pomdps. *14èmes Journées Francophones sur les Systèmes Multi-Agents - JFSMA'2006*.