Fonctions de Valeurs Distribuées sous Contraintes de Communication pour la Coordination Décentralisée d'Agents Décisionnels

Laëtitia Matignon, Laurent Jeanpierre, Abdel-Illah Mouaddib

Fonctions de Valeurs Distribuées sous Contraintes de Communication pour la Coordination Décentralisée d'Agents Décisionnels ⋆

Laëtitia Matignon, Laurent Jeanpierre and Abdel-Illah Mouaddib

Université de Caen Basse-Normandie, CNRS - UMR6072 GREYC - F-14032 Caen, France
{laetitia.matignon,laurent.jeanpierre,abdel-illah.mouaddib}@unicaen.fr

Résumé : De récents travaux relatifs aux modèles décisionnels exploitent les interactions locales et la coordination pour résoudre les processus décisionnels de Markov partiellement observables. Dans cet article, nous proposons une approche basée sur ces techniques de résolution orientée-interaction et motivée par une application réelle d'agents décisionnels décentralisés pour l'exploration et la cartographie d'un environnement inconnu. Cette approche utilise des fonctions de valeurs distribuées (DVF) qui découplent le problème multi-agent en un ensemble de problèmes individuels et considèrent les interactions possibles entre agents dans une classe de résolution séparée. Cela permet une réduction significative de la complexité de résolution du processus décisionnel de Markov décentralisé en le résolvant comme une collection de processus décisionnels de Markov. Cependant les techniques de DVF existantes supposent une communication entre les agents permanente et gratuite. Dans cet article, nous étendons la méthode par DVF de sorte à ce qu'elle prenne en compte une observabilité locale totale, un partage limité d'information entre les agents et des coupures possibles de communication. Nous appliquons notre nouvelle DVF à une application réelle consistant en l'exploration multi-robot d'un environnement inconnu où chaque robot calcule localement une stratégie qui minimise les interactions entre les robots et maximise la couverture de l'espace par l'équipe même sous contraintes de communication. Notre technique a été implémentée et évaluée en simulation et sur des scénarios réels lors d'un défi robotique pour l'exploration et la cartographie d'un environnement inconnu. Nous présentons des résultats expérimentaux issus de scénarios réels et du défi où notre système est arrivé vice-champion.

1 Introduction

Les récentes avancées concernant la résolution des processus décisionnels de Markov partiellement observables (Dec-POMDPs) ont permis une augmentation notable de la taille des problèmes pouvant être résolus. En particulier, une des directions les plus prometteuses est de tirer parti des interactions locales avec une résolution orientée-interaction (OI) (Canu & Mouaddib, 2011; Melo & Veloso, 2011; Velagapudi *et al.*, 2011). De telles approches relâchent l'hypothèse très restrictive de dépendances totales posée dans les Dec-POMDPs. Cette hypothèse considère que les agents sont en permanence en interaction. Ces approches de résolution orientée-interaction ouvrent des perspectives prometteuses concernant des applications réelles d'agents décisionnels décentralisés.

L'approche développée dans cet article est motivée par l'utilisation des Dec-POMDPs résolus par des techniques orientée-interaction (OI) pour l'exploration et la cartographie d'un environnement inconnu par une flotte de robots mobiles. Ce système a été développé et appliqué avec succès à des scénarios réels au cours d'un défi robotique [1] DGA [2]/ANR [3] pour l'exploration, la cartographie et la reconnaissance d'objets par des robots mobiles. Dans cet article, l'aspect SLAM [4] distribué n'est pas considéré et nous nous

⋆. Ces travaux sont subventionnés par l'Agence Nationale de Recherche (ANR) et la Direction générale de l'armement (DGA) à travers le projet ANR-09-CORD-103. Ils sont développés conjointement avec les membres du consortium Robots_Malins.

1. http ://www.defi-carotte.fr/
2. Direction générale de l'armement.
3. Agence Nationale de Recherche.
4. *Simultaneous Localization And Mapping*.

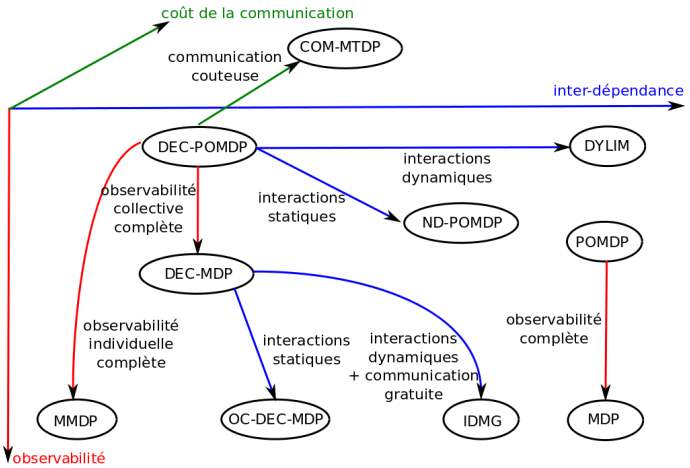


FIGURE 1 – Taxonomie des modèles de décision selon l'indépendance, la communication et l'observabilité.

intéressons uniquement au modèle décisionnel. Nous considérons que les robots sont indépendants et peuvent partager des informations limitées en communiquant, permettant ainsi de compléter leur observabilité. Cependant les contraintes de communication sont un défi majeur étant donné que des coupures potentielles de communication peuvent survenir et mener à une perte des informations qui sont partagées entre les robots.

Cet article s'intéresse à la coordination globale et locale d'agents décisionnels décentralisés sous les hypothèses d'observabilité locale totale, de partage limité d'information entre les agents et de coupures possibles de communication. La coordination globale est définie comme comme l'allocation appropriée de buts pour les robots individuels et la minimisation des interactions qui mènent à des conflits entre les membres de l'équipe. La coordination locale consiste en la résolution des situations d'interaction locale. Les hypothèses que nous avons posées ne sont pas considérées simultanément dans la littérature. Par exemple, les méthodes de résolution OI ne prennent pas en compte toutes ces hypothèses ; les techniques classiques de négociation supposent une communication permanente ; et les approches existantes d'exploration multi-robot considérant des contraintes de communication gèrent seulement les problèmes de portée de communication limitée et ne sont pas applicables dans le cas de coupures stochastiques de communication. Nous introduisons donc dans cet article une nouvelle méthode de résolution OI pour des modèles décisionnels décentralisés dans le cas de partage limité d'information et de coupures de communication.

Par la suite, nous présentons tout d'abord les travaux relatifs à la résolution OI des Dec-POMDPs et à l'exploration multi-robot puis le contexte de nos travaux. Ensuite nous introduisons l'approche DVF et son extension pour supporter les coupures de communication. L'application de ce modèle à l'exploration multi-robot est détaillée. Finalement, des expériences issues de scénarios réels et de simulations sont présentées afin de montrer l'efficacité de notre approche lors de coupures potentielles de communication.

2 Travaux Relatifs

2.1 Résolution Orientée-Interaction des Dec-POMDPs

La plupart des travaux relatifs aux Dec-POMDPs cassent la complexité de résolution de ces modèles en posant des hypothèses selon une ou plusieurs des directions suivantes : l'*observabilité* selon que chaque agent a une connaissance partielle ou complète de l'état du monde ; la *communication* selon la possibilité et le coût du partage d'information entre les agents ; et l'*inter-dépendance* qui exploite la structure du problème comme la localité des interactions, la décomposition des récompenses et l'indépendance entre les agents. Toutes les approches existantes de résolution des Dec-POMDPs vérifient telles ou telles hypothèses parmi celles-ci. La figure 1 regroupe certains de ces modèles et les liens classiques entre les différents modèles. Les approches mono-agent (MDP et POMDP) se différencient selon l'hypothèse d'observabilité complète ou partielle de l'agent.

Concernant les modèles multi-agent et l'observabilité, le modèle Dec-POMDP revient à un modèle Dec-MDP quand l'observabilité collective est complète et à un modèle MMDP quand l'observabilité individuelle est totale. De récents travaux prometteurs exploitent l'hypothèse d'inter-dépendance avec une résolution OI des Dec-POMDPs. Ces travaux exploitent la structure du problème et ont abouti à de nouvelles classes de modèles basés sur un ensemble de modèles de décision individuels en interaction. Ainsi la complexité de résolution des Dec-POMDPs est réduite. Le modèle ND-POMDP (Nair *et al.*, 2005) suppose une structure d'interaction statique, c'est-à-dire qu'un agent interagit toujours avec le même sous-ensemble d'agents. Dans le cas d'une observabilité collective totale et d'une structure d'interaction statique, le modèle Dec-MDP est réduit à un OC-Dec-MDP (Beynier & Mouaddib, 2005, 2006). Néanmoins, interagir tout le temps avec les mêmes agents n'est pas réaliste. Des modèles ont alors été proposés supposant des interactions dynamiques telles qu'un agent interagit avec un ensemble dynamique d'agents : IDML (Spaan & Melo, 2008) et DyLIM (Canu & Mouaddib, 2011).

Parmi tous ces modèles, aucune communication explicite n'est supposée excepté pour le modèle IDMG qui présume des communications illimitées et gratuites entre les groupes d'agents interagissant ensemble. Peu de modèles étudient le coût de la communication comme le modèle COM-MTDP (Pynadath & Tambe, 2002).

2.2 Exploration Multi-Robot

L'exploration multi-robot a reçu une attention considérable ces dernières années. Des stratégies d'exploration variées ont été proposées qui diffèrent principalement par la manière dont la coordination globale est réalisée. Dans (Burgard *et al.*, 2005; Wurm *et al.*, 2008), elle est centralisée. L'utilité de chaque cible est calculée comme un compromis entre le gain espéré à cette cible (zone explorée espérée quand la cible sera atteinte prenant en compte les recouvrements possibles des capteurs des robots) et le coût pour atteindre cette cible. La coordination globale est réalisée en assignant différentes cibles aux robots, maximisant ainsi la couverture de l'espace et réduisant les recouvrements entre les zones explorées par chaque robot. La coordination globale peut aussi être décentralisée comme dans (Bautin *et al.*, 2011) où une frontière d'exploration est allouée à chaque robot de sorte à ce qu'il y ait le moins de robots entre cette frontière et le robot à qui elle a été assignée. Dans (Zlot *et al.*, 2002) les robots enchérissent sur les cibles pour négocier leurs allocations.

La plupart des approches supposent que les robots maintiennent une communication permanente pendant l'exploration afin de partager les informations qu'ils récupèrent ainsi que leurs positions. Cependant, une communication gratuite et permanente est rarement le cas en pratique et un défi majeur est la prise en compte de coupures potentielles de communication. Quelques approches récentes considèrent la contrainte de portée de communication limitée. (Burgard *et al.*, 2005) appliquent leur stratégie d'exploration multi-robot à chaque sous-système de robots capables de communiquer entre eux. Ceci résulte dans le pire des cas en une situation où tous les robots explorent individuellement tout l'environnement. (Powers *et al.*, 2004) proposent une méthode pour maintenir la connectivité de l'équipe pendant l'exploration de sorte à ce que les robots restent en permanence à portée de communication de l'un d'entre eux. (Hoog *et al.*, 2010) imposent des contraintes périodiques de rendez-vous où les robots doivent se retrouver à portée des autres afin de partager des informations.

3 Contexte

3.1 POMDPs Décentralisés

Le modèle Dec-POMDP (Bernstein *et al.*, 2002) permet la représentation de problèmes de décision de type MDP, lorsque ceux-ci sont à la fois multi-agent et en environnement partiellement observable. Un Dec-POMDP est défini par un tuple $< I, S, A, T, R, \Omega, O >$. I est le nombre d'agents, S un ensemble d'états joints and $A = \{A_i\}$ un ensemble d'actions jointes. Un état joint du problème peut être décomposé en $s = (s_1, ..., s_I)$ tel que s_j est l'état du robot j. A_i définit l'ensemble des actions individuelles a_i du robot i. $T : S \times A \times S \rightarrow [0; 1]$ est une fonction de transition et $R : S \times A \rightarrow \Re$ une fonction de récompense. Ω est un ensemble d'observations qu'un agent peut recevoir sur un environnement et $O : S \times A \times S \times \Omega \rightarrow [0; 1]$ une fonction d'observation. Si l'état global du système est collectivement totalement observable, le modèle Dec-POMDP est réduit à un modèle Dec-MDP.

Un modèle décisionnel de Markov (MDP) (Puterman, 1994) peut être vu comme un Dec-MDP à un agent où $I = 1$. Un MDP est défini par un tuple $< S, A, T, R >$. L'objectif de la planification est alors de trouver une séquence d'actions maximisant la récompense espérée sur le long terme. Un tel plan est appelé politique $\pi : S \rightarrow A$. Une politique optimale π^* spécifie pour chaque état s l'action optimale à exécuter à l'instant courant en supposant que l'agent agira de manière optimale dans le futur. La valeur de π^* est donnée par la fonction de valeur optimale V^* satisfaisant l'équation d'optimalité de Bellman :

$$V^*(s) = \max_{a \in A}(R(s,a) + \gamma \sum_{s' \in S} T(s,a,s')V^*(s')) \quad (1)$$

où γ est un coefficient d'atténuation.

Pour résoudre un Dec-POMDP, une politique jointe optimale est calculée. Cependant la complexité de résolution d'un Dec-POMDP est de classe NEXP-complet (Bernstein *et al.*, 2002). Calculer une politique optimale pour un Dec-POMDP est donc en général considéré comme très difficile. Les récentes méthodes de résolution OI présentées dans la partie 2.1 permettent de réduire cette complexité.

3.2 Fonctions de Valeurs Distribuées

Les fonctions de valeurs distribuées (DVF) ont été introduites par Schneider *et al.* (1999) pour distribuer l'apprentissage par renforcement parmi différents agents dans le cas de systèmes distribués, tels que les environnements multi-robot avec un couplage faible entre les agents. Dans Babvey *et al.* (2003), cette aproche distribuée permet à des robots d'apprendre des comportements concurrentiels dans des scénarios de navigation multi-robot.

La DVF pose plusieurs hypothèses sur le modèle. L'état doit être globalement totalement observable et la fonction de récompense R décomposable en une somme ($R = \sum_{i \in I} R_i$) où chaque terme R_i peut être calculé directement à partir de l'état global du système et de l'action individuelle d'un agent ($R_i : S \times A_i \rightarrow \Re$). Afin de coordonner les actions individuelles des agents, la DVF est calculée par chaque agent à partir d'informations échangées entre les agents voisins sur leurs fonctions de valeur. La fonction de valeur distribuée $V_i(s)$ pour un agent i dans l'état $s \in S$ est définie par :

$$V_i(s) = \max_{a \in A_i}(R_i(s,a) + \gamma \sum_{j \in I} f_{ij} \sum_{s' \in S} T(s,a,s')V_j(s')) \quad (2)$$

où f_{ij} est un coefficient de pondération constant qui détermine de quelle manière un agent j pondérera la fonction de valeur distribuée d'un agent i. En particulier, f_{ij} est mis à 0 pour tous les agents j non-voisins de i. La nature récursive de la fonction de valeur distribuée permet à des agents non-voisins d'interagir. Cependant, les approches utilisant les DVFs posent une hypothèse forte de communication permanente et gratuite.

4 Fonctions de Valeurs Distribuées sous Contraintes de Communication

Après une présentation de nos motivations et des hypothèses posées pour ces travaux, nous introduisons l'approche DVF et son extension pour la prise en compte de coupures de communication. Ensuite quelques problèmes sont discutés.

4.1 Motivations

Notre motivation première est la coordination décentralisée d'agents décisionnels sous les hypothèses de partage limité d'information entre les agents et de coupures potentielles de communication. Nous nous intéressons à la coordination globale d'agents définie comme l'allocation appropriée de buts pour les agents individuels et la minimisation des interactions pouvant mener à des conflits entre les différents agents de l'équipe. La coordination locale est aussi nécessaire afin de résoudre les situations d'interaction locale.

Dans cette perspective, notre approche se place dans le cadre des Dec-(PO)MDPs pour la modélisation d'équipes coopératives d'agents décisionnels décentralisés. Afin de réduire la complexité de résolution

de ce modèle, nous utilisons une approche de résolution orientée-interaction (OI). Ainsi le modèle Dec-POMDP peut être décrit avec deux classes : une classe sans interaction représentée comme un ensemble de (PO)MDPs, un par agent ; et une classe d'interaction pour les interactions proches ou locales. Notre approche de résolution OI est basée sur la technique des DVFs. Elle permet la coordination des agents par un partage limité d'information entre les agents. En effet, la technique par DVF résout le modèle Dec-(PO)MDP comme une collection de (PO)MDPs : chaque agent calcule individuellement sa DVF en considérant les interactions entre les (PO)MDPs par un partage d'information entre les agents ou en inferrant ces informations par empathie. Ainsi chaque agent calcule localement une stratégie qui minimise les conflits, *i.e.* qui évite aux agents d'entrer dans la classe d'interaction. La classe d'interaction est une classe séparée résolue indépendemment en calculant les politiques jointes pour ces états joints spécifiques. Afin de prendre en compte les coupures potentielles de communication, la technique DVF est améliorée pour relâcher les hypothèses de communication gratuite et permanente.

4.2 Hypothèses

Nous supposons que l'observabilité locale de chaque agent est totale donc notre approche se place dans le cadre des modèles Dec-MDPs. En utilisant une méthode de résolution OI, le modèle Dec-MDP est résolu comme un ensemble de MDPs locaux à chaque agent. Un MDP local à un agent $i \in I$ est défini par $MDP_i =< S_i, A_i, T_i, R_i >$. Nous supposons que nos agents sont homogènes, ils ont donc le même espace d'états S_i et d'actions A_i. Nous supposons aussi que chaque agent communique aux autres uniquement son état courant. Donc si la communication est permanente, l'agent i connaît à chaque pas de temps t l'état $s_{ij} \in S_i$ de chaque autre agent j. Si la communication est coupée, les états des autres agents ne sont plus connus à chaque pas de temps. Nous supposons alors que $s_{ij} \in S_i$ est le dernier état connu (reçu) d'un autre agent j à l'instant t_j. C'est-à-dire que t_j est le dernier pas de temps où la communication était effective entre les agents i et j. A l'instant courant t, l'agent i sait que l'agent j était dans l'état s_{ij} il y a $\Delta t_j = t - t_j$ pas de temps.

4.3 Fonctions de Valeurs Distribuées

Dans un article récent (Matignon *et al.*, 2012b), nous avons formalisé la résolution orientée interaction des Dec-MDPs avec des fonctions de valeurs distribuées (DVFs). Dans ces travaux, la technique de résolution basée sur les DVFs permet à chaque agent de choisir un but qui ne sera pas choisi par les autres. Chaque agent décide individuellement le but à choisir en considérant les buts pouvant être sélectionnés par les autres agents. La valeur d'un but dépend des récompenses espérées à ce but et du fait qu'il ne sera probablement pas choisi par les autres. Nous reformulons la fonction de valeur distribuée originale (eq. 2). La DVF est définie localement par chaque agent i dans son MDP_i local $< S_i, A_i, T_i, R_i >$. Dans le cas de communication permanente, un agent i calcule sa DVF V_i selon :

$$\forall s_i \in S_i \quad V_i(s_i) = \max_{a_i \in A_i} \left(R_i(s_i, a_i) + \gamma \sum_{s' \in S_i} T_i(s_i, a_i, s') \; [V_i(s') - \sum_{j \neq i} f_{ij} P_r(s'|s_{ij}) V_j(s')] \right) \quad (3)$$

où $P_r(s'|s_{ij})$ est la probabilité pour un agent j d'atteindre l'état s' depuis l'état s_{ij} et f_{ij} est un coefficient de pondération qui détermine de quelle manière la fonction de valeur de l'agent j réduit celle de l'agent i.

4.4 Fonctions de Valeurs Distribuées sous Contraintes de Communication

Étant données nos contraintes sur la communication, nous relâchons les hypothèses fortes qui sont faites concernant la communication dans la méthode par DVF présentée précédemment (§4.3). Tout d'abord, les agents ne peuvent pas échanger d'information concernant leurs fonctions de valeur (partage de V_j). Cependant, chaque agent i peut calculer la fonction de valeur de chacun des autres agents V_j par empathie. Ensuite, notre approche doit être robuste face aux coupures de communication, *i.e.* que nous considérons que les états des autres agents s_{ij} ne sont pas toujours connus à chaque pas de temps par un agent i. Sous ces hypothèses plus restrictives de partage limité d'information et de coupures de communication, l'équation 3

peut être réécrite :

$$\forall s_i \in S_i \quad V_i(s_i) = \max_{a_i \in A_i} \left(R_i(s_i, a_i) + \gamma \sum_{s' \in S_i} T_i(s_i, a_i, s') \ [V_i(s') - \sum_{j \neq i} f_{ij} P_r(s'|s_{ij}, \Delta t_j) V_j(s')] \right) \tag{4}$$

où $P_r(s'|s_{ij}, \Delta t_j)$ est la probabilité que l'agent j transitionne dans l'état s' sachant qu'il était dans l'état s_{ij} il y a Δt_j pas de temps.

Si la communication est permanente, le dernier état connu d'un autre agent j est son état à l'instant courant $t_j = t$ donc $\Delta t_j = 0$. Nous posons :

$$P_r(s'|s_{ij}, \Delta t_j = 0) = P_r(s'|s_{ij}) \tag{5}$$

Ainsi l'équation 4 est une extension de la fonction de valeur distribuée dans le cas de coupures de communication potentielles.

Cependant la difficulté à calculer cette DVF réside dans l'estimation par un agent i de la probabilité de transition pour un autre agent j sachant que cet autre agent était dans l'état s_{ij} il y a Δt_j pas de temps. Cette probabilité pourrait être évaluée selon :

$$P_r(s'|s_{ij}, \Delta t_j) = \eta \sum_{\tau \in Traj(s_{ij}, s')} P_r(\tau, \Delta t_j) \tag{6}$$

où η est une constante de normalisation, $Traj(s_{ij}, s')$ est l'ensemble des trajectoires d'un état s_{ij} à s' et $P_r(\tau, \Delta t_j)$ est la probabilité que l'agent j suive la trajectoire τ. La somme sur toutes les trajectoires possibles est la croyance que l'agent j soit à l'état s' sachant qu'il était en s_{ij} il y a Δt_j pas de temps.

La complexité de calcul de cette probabilité dépend de plusieurs facteurs. Tout d'abord, selon le modèle utilisé, calculer l'ensemble des trajectoires d'un état à un autre dans un MDP peut rapidement devenir très complexe. Cependant, cette complexité pourrait être réduite en utilisant des modèles structurés, tels que les diagrammes de Voronoi dans le cas d'une application robotique. De plus, l'application visée peut aussi mener à une simplification du calcul de cette probabilité, comme par exemple dans le cas d'une application d'exploration détaillée dans la section suivante.

4.5 Coordination Locale

Chaque agent calcule sa stratégie d'exploration avec une DVF de sorte à minimiser les interactions. Néanmoins si des situations d'interaction locale surviennent, la DVF ne permet pas de les résoudre et la coordination locale doit alors être résolue avec une autre technique. Par exemple les politiques jointes associées aux états joints spécifiques à ces interactions proches pourraient être calculées off-line.

Ainsi, la DVF permet de réaliser la coordination décentralisée d'agents décisionnels sous la contrainte de coupures de communication. Cependant, son applicabilité reste limitée par la complexité du calcul de la probabilité de transition. Dans la section suivante, cette complexité est réduite dans le contexte d'une application d'exploration multi-robot.

5 Exploration Multi-Robot

Notre approche est motivée en premier lieu par une application robotique d'exploration d'un environnement inconnu par une flotte de robots mobiles sous contraintes de communication. Nous considérons que l'état de chaque robot est connu à chaque instant de décision. De plus, lorsque les robots peuvent communiquer, chaque robot a accès à une carte locale mise à jour avec les zones explorées par tous les robots de la flotte et aux positions des autres robots dans cette carte.

Notre approche utilise le formalisme Dec-MDP résolu comme un ensemble de modèles MDPs par la technique par DVF. Une DVF est calculée localement par chaque robot. Ainsi chaque robot considère les

zones qui pourraient être explorées par les autres et réduit d'autant son intérêt pour ces zones. En effet, la DVF spécifie que le gain espéré par un robot pour explorer une zone est réduit par ce que les autres peuvent espérer gagner en visitant cette même zone. Le robot suivant une politique calculée *via* la DVF va alors choisir l'action avec le gain espéré maximal donc l'action pour laquelle les intérêts des autres pourraient être faibles. En d'autres termes, en calculant localement une DVF et en suivant la politique résultante, les robots choisissent des actions qui minimisent leurs interactions locales (pouvant mener à des conflits comme explorer les mêmes zones de l'environnement) tout en maximisant la couverture de l'espace. Néanmoins, l'application de cette méthode dans le cas de coupures de communication est complexe. Nous montrons dans cette section comment cette complexité de calcul peut être réduite dans le contexte de l'exploration multi-robot.

5.1 Fonctions de Valeurs Distribuées pour l'Exploration Multi-Robot

Dans le cas de coupures de communication, l'équation de la DVF (eq. 4) est difficilement applicable étant donnée la complexité de calcul de la probabilité de transition $P_r(s'|s_{ij}, \Delta t_j)$. D'une manière similaire à (Burgard *et al.*, 2005), nous pouvons considérer qu'un robot ignore les informations sur les membres de l'équipe avec lesquels la communication a été perdue *i.e.* :

$$\forall s_i \in S_i \quad V_i(s_i) = \max_{a_i \in A_i} \left(R_i(s_i, a_i) + \gamma \sum_{s' \in S_i} T_i(s_i, a_i, s') \ [V_i(s') - \sum_{j \in \Omega(i)} f_{ij} P_r(s'|s_{ij}) V_j(s')] \right) \quad (7)$$

où $\Omega(i)$ est pour le robot i l'ensemble des robots avec lesquels il est toujours en communication. Évidemment, dans le pire des cas, on aura une situation où tous les robots vont agir comme s'ils étaient indépendants et ainsi explorer individuellement tout l'environnement.

Dans le cas de notre application d'exploration, la complexité du calcul de la probabilité de transition $P_r(s'|s_{ij}, \Delta t_j)$ peut être réduite. En effet l'estimation de cette probabilité (eq. 6) peut être réécrite selon :

$$P_r(s'|s_{ij}, \Delta t_j) = \eta \sum_{s'' \in S_i} P_r(s'|s'') P_r(s''|s_{ij}, \Delta t_j - 1) \quad (8)$$

Afin d'évaluer successivement chaque probabilité d'un état à l'autre $P_r(s'|s'')$ jusqu'à ce que l'état s_{ij} soit atteint, un algorithme de propagation "par vague" peut être appliqué à partir du dernier état connu s_{ij} du robot j. Les valeurs obtenues peuvent être normalisées de sorte à être consistante avec des valeurs de probabilités. Ces valeurs représentent alors les intentions futures d'un robot depuis son dernier état connu. Cependant cette approximation aura une efficacité d'autant plus réduite que la communication est coupée depuis un long moment.

5.2 Algorithme

Les différentes étapes d'une boucle de décision basée sur le calcul de la DVF et dans le cas de l'exploration sont données à l'Algorithme 1 pour un agent i. Une boucle de décision consiste en la mise à jour du modèle MDP local de l'agent à partir des nouvelles données récupérées (données d'exploration et de position des autres). Ensuite, selon la classe dans laquelle est l'agent i (classe d'interaction ou sans interaction), une politique est suivie. Plus de détails sur le modèle MDP local sont donnés au paragraphe 6.1.

5.2.1 Classe Sans Interaction et DVF

Les robots que nous considérons sont supposés homogènes donc les fonctions de valeur V_j des autres robots j peuvent être calculées en une seule fois par le calcul de la fonction de valeur empathique V_{emp} par l'algorithme d'itération sur les valeurs (Bellman, 1957). Pour évaluer la probabilité de transition d'un autre robot j, le robot i applique un algorithme de propagation par vague à partir du dernier état connu s_j du robot j. Le coefficient de pondération f_{ij} permet d'équilibrer la fonction de valeur en fonction des récompenses.

Les robots planifient en continu et mettent à jour leurs modèles locaux et politiques selon leur perception des changements de l'environnement. Cela permet de mettre à jour rapidement leur plan d'action afin de réagir aussi rapidement que possible aux décisions des autres et aux informations récoltées en route [5].

5. La carte est souvent explorée avant que le robot n'est atteint sa cible.

Algorithme 1 : Pseudo-code de l'exploration pour un robot i utilisant la nouvelle DVF

```
/*VI est l'algorithme d'itération sur les valeurs                    */
/*WP est l'algorithme de propagation par vague                       */
début
    tant que l'exploration n'est pas finie faire
        /*Une boucle de décision                                     */
        Récupération des nouvelles données d'exploration
        Récupération et communication de son état courant s_i
        Mise à jour du modèle MDP local MDP_i =< S_i, A_i, T_i, R_i >
        pour tous les j ∈ Ω(i) faire
            Réception de s_ij
            Calcul de la distance aux autres robots dist(s_i, s_ij)
        si ∃k tel que dist(s_i, s_k) < δ alors
            /*Classe d'interaction et coordination locale            */
            Suivre la politique jointe calculée off-line
        sinon
            /*Classe sans interaction et DVF                         */
            V_emp ← VI (MDP_i, γ)
            pour tous les j ≠ i faire
                P_r(*|s_ij, Δt_j) ← WP (s_ij)
                V_j = V_emp
                f_ij = max_s R_i(s) / max_s V_j(s)
            V_i ← DVF (MDP_i, f_ij, γ, V_j, P_r)
            Suivre la politique associée à V_i
fin
```

Cependant, cela nécessite que la boucle de décision soit assez rapide pour une utilisation en-ligne. Étant donnée que le modèle est mis à jour à chaque boucle de décision, nous utilisons une approche gloutonne qui planifie sur un horizon à court terme.

5.2.2 Classe d'Interaction et Coordination locale

La nouvelle DVF permet le calcul de stratégies qui minimisent les interactions locales entre les agents (ou robots). Cependant elle ne résout pas les situations où les robots sont proches ce qui peut par exemple arriver dans la zone de départ [6] ou dans des couloirs étroits. Afin de résoudre la coordination locale nécessaire dans ces situations, un MDP multi-agent (MMDP) (Boutilier, 1996) peut être utilisé quand les robots concernés par l'interaction sont proches les uns des autres. Les politiques jointes peuvent alors être calculées hors-ligne pour ces états joints spécifiques et suivies quand la coordination locale doit être appliquée. Dans notre contexte applicatif, ces situations peuvent être facilement reconnues en calculant la distance entre un robot et ses partenaires. Quand cette distance est inférieure à un seuil de sécurité prédéfini, la coordination locale est appliquée et résulte en différents comportements selon la position des robots les uns par rapport aux autres. Dans le cas où un robot suit un autre robot ou s'ils sont face à face, un robot stoppe et attend que l'autre bouge et s'éloigne. Un seuil de sécurité d'urgence défini un mouvement de marche arrière de sorte à ce qu'un robot recule pour laisser passer l'autre. Si la communication est perdue, la coordination locale ne peut pas être appliquée car elle est non-détectée mais le système de bas-niveau d'anti-collision permet d'éviter une collision entre les robots. De telles situations sont illustrées dans nos expériences.

6. Dans la cadre du défi robotique, tous les robots doivent démarrer et retourner dans une zone spécifique où les interactions locales surviennent nécessairement.

6 Expériences

Le modèle décisionnel de notre système d'exploration multi-robot est basé sur un ensemble de MDPs locaux à chaque robot. Chaque MDP est résolu indépendamment par chaque robot en calculant la DVF. Nous détaillons tout d'abord les composants du modèle MDP local. Ensuite les plateformes expérimentales sont décrites. Elles sont de deux types : une plateforme robotique composée de deux robots mobiles et une plateforme de simulation avec le simulateur Stage.

6.1 Modèle MDP Local

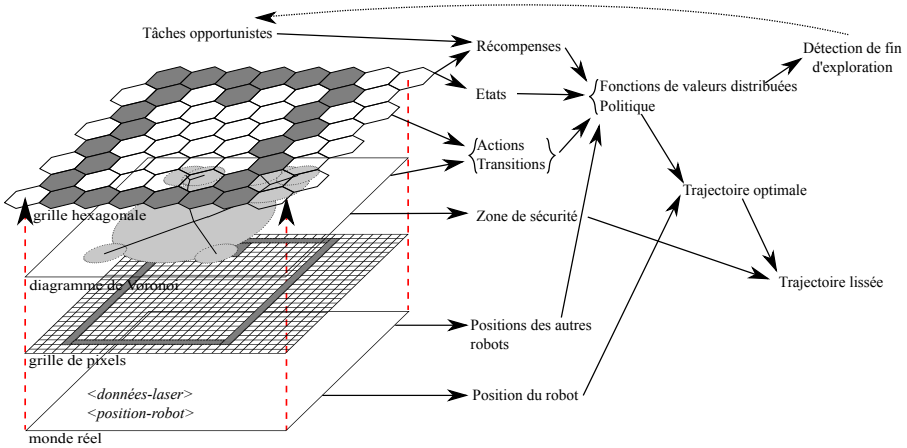
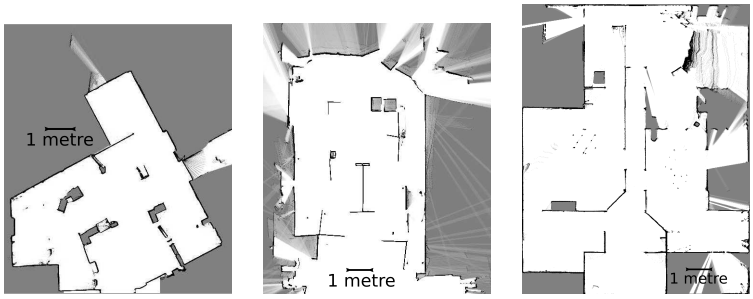


FIGURE 2 – Architecture globale du modèle décisionnel local à un robot

La Figure 2 décrit l'architecture globale du modèle décisionnel local à un robot. Il est composé d'une grille à 4 niveaux. Le premier niveau est le monde réel où les robots se déplacent. Le second est la grille de pixels. Chaque cellule de la grille est initialisée comme inconnue et mise à jour avec la valeur libre (pas d'obstacles) ou occupée (présence d'obstacle) par le processus d'acquisition de données. Le diagramme de Voronoi et la grille hexagonale sont chacun calculés à partir de la grille de pixels et utilisés pour générer les structures de données du modèle MDP local.

Les états et récompenses sont basés sur la grille hexagonale. Chaque hexagone est composé d'un ensemble de pixels et est considéré comme inconnu, libre ou occupé selon la valeur de ses pixels. Un *état* du MDP local est la position hexagonale du robot dans la grille hexagonale (son hexagone et son orientation parmi les 6 orientations possibles). La fonction de récompense d'exploration est calculée par un mécanisme de propagation des récompenses basé sur le gain d'information espéré dans chaque hexagone comme détaillé dans (Gloannec *et al.*, 2010). Les *actions* du robot modélisent le mouvement de celui-ci : avancer, tourner à droite ou à gauche, et attendre. Afin d'éviter les mouvements de *zig-zag* pour se déplacer d'un hexagone à l'autre qui apparaissent dans les espaces encombrés ou étroits, une action supplémentaire est modélisée consistant à suivre une arrête de Voronoi jusqu'au prochain hexagone. Concernant *la fonction de transition*, si l'action entreprise est autorisée par la zone de sécurité calculée par le diagramme de Voronoi, l'état espéré est atteint avec une forte probabilité. Sinon l'état n'est pas modifié et une forte pénalité est ajoutée à la fonction de récompense pour ce couple état-action. Les mouvements autorisés ne sont pas toujours déterministes car une zone de sécurité faible peut ralentir le robot, ou même entraîner un échec du mouvement désiré. Enfin, la trajectoire optimale hexagonale est lissée en agrégeant les suites d'actions en série de points de navigation respectant les zones de sécurité.



(a) Expériences en laboratoire. (b) Défi robotique.

FIGURE 3 – Cartes de pixels obtenues après exploration avec deux robots. La gamme de couleur des pixels s'étend de noir (obstacle) à blanc (libre).

6.2 Plateformes Expérimentales

6.2.1 Robots Réels

Nous disposons de μ-troopers Wifibot [7] qui sont des robots mobiles à 6 roues dotés d'un processeur Intel Core 2 Duo, 2GB RAM et 4GB flash. Chaque robot est équipé de télémètres laser. L'architecture logicielle est centrée autour du modèle de distribution de données publisher/subscriber *Data Distribution System* (DDS) [8]. Ce réseau d'échange d'information permet aux différents programmes de tourner en parallèle, même sur différents ordinateurs. Dans notre architecture, différents modules peuvent tourner de manière asynchrone : l'acquisition laser, le SLAM, la décision et la mobilité. Chaque robot est indépendant et possède ses propres modules. Le module SLAM, basé sur (Xie *et al.*, 2010), reçoit les données laser et fournit aux autres modules la position du robot. L'architecture permet aussi aux robots d'échanger leurs données laser et leur position. Tant que la communication n'échoue pas, chaque robot connaît les zones explorées par les autres et met à jour sa carte locale avec les données de son laser local et des lasers distants. Pendant une coupure de communication, rien n'est échangé entre les robots. Cependant, notre architecture est robuste aux coupures de communication. Dès que la communication est rétablie, la carte locale de chaque robot est mise à jour avec les zones explorées par les autres et leurs positions relatives sont à nouveau échangées. Le module de mobilité implémente des algorithmes avancés de lissage de trajectoire et de suivi de chemin, ainsi qu'une fonction de retour sur trace pour prévenir les situations de blocage du robot. Le module de décision tourne de manière asynchrone et calcule une nouvelle politique en moyenne chaque seconde selon l'algorithme détaillé à l'Algorithme 1.

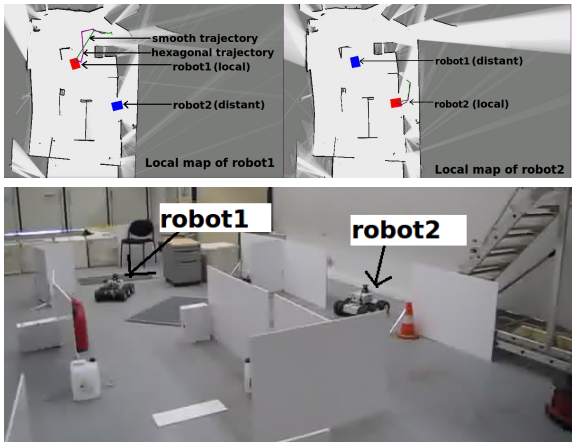
6.2.2 Robots Simulés

Nous utilisons le simulateur 2D Stage [9] avec une architecture qui imite celle des robots réels. DDS est remplacé par des communications inter processus basées sur des mécanismes de mémoire partagée. L'acquisition laser est simulée par un capteur "laser" virtuel. Un capteur de position virtuel simule à la fois le module SLAM en donnant des données odométriques et le module de mobilité en exécutant un algorithme de suivi de chemin. Enfin le module décision utilisé sur les robots est interfacé sans modifications avec le simulateur 2D.

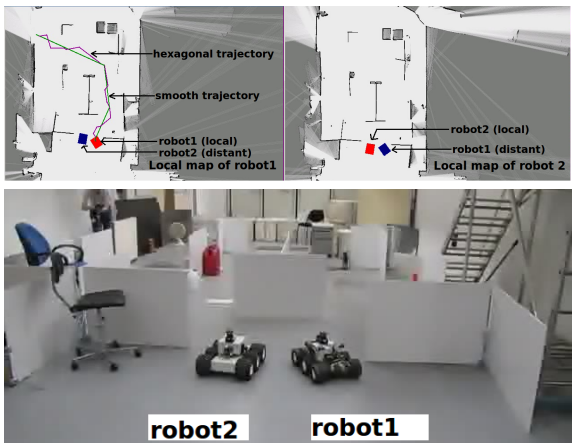
6.3 Résultats Expérimentaux

Nous avons montré que la technique basée sur la DVF permet de coordonner une équipe de robots pour l'exploration sans coupures de communication (Matignon *et al.*, 2012a,b). Dans ce paragraphe, nous présentons des résultats issus de scénarios en simulation et avec des robots réels où nous testons différents schémas de communication (communication permanente, pas de communication et coupures de communication) afin de montrer l'efficacité de la nouvelle DVF quand la communication n'est pas fiable.

7. www.wifibot.com
8. http ://www.opensplice.com
9. http ://playerstage.sourceforge.net/



(a) Coordination globale.



(b) Coordination locale.

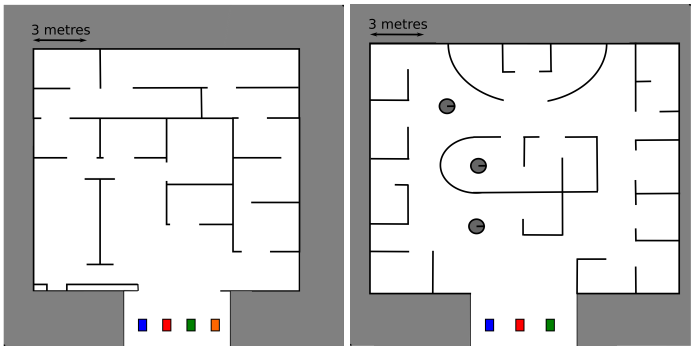
FIGURE 4 – Aperçu d'une mission d'exploration et cartes de pixels locales à chaque robot.

6.3.1 Robots Réels

Nous avons réalisé des expériences avec deux μ-troopers. Les vidéos, disponibles à `http://lmatigno.perso.info.unicaen.fr/research`, montrent différentes missions d'exploration des robots. Les situations intéressantes sont mises en évidence (répartition globale de la tâche, coordination locale, retour à la zone de départ en fin de mission). Les cartes de pixels obtenues en fin de missions sont à la figure 3a. La figure 3b est la carte de pixels issue du défi robotique de l'année 2011. La figure 4a montre les cartes de pixels locales à chaque robot au cours d'une mission d'exploration. Les trajectoires planifiées par chaque robot sont tracées sur ces cartes. Le robot distant est vu par le robot local dans sa carte locale. A ce moment de la mission, les robots se partagent la zone à explorer : le robot 1 décide d'explorer le haut de la carte et le robot 2 d'explorer un couloir en haut à droite de la carte. La figure 4b illustre une situation de coordination locale résolue avec succès : le robot 1 décide de se déplacer vers la droite alors que le robot 2 attend que l'autre robot passe la porte. Aucune coupure de communication ne s'est produite au cours de ces tests. Néanmoins, nous avons observé des coupures de communication temporaires pendant d'autres tests et nous avons pu remarquer que notre approche permettait une exploration efficace tout en étant robuste aux coupures de communication. En effet, une fois la communication rétablie, les robots partagent à nouveau leurs données laser et positions grâce à notre architecture robuste aux coupures.

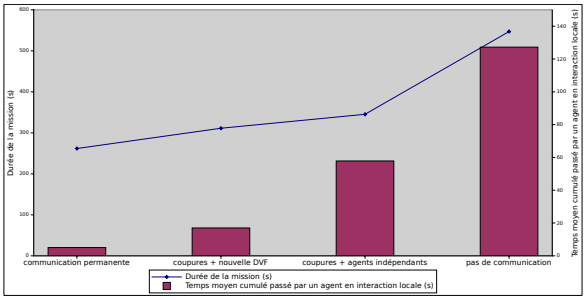
6.3.2 Robots Simulés

Environnements de simulation denses Afin de montrer l'intérêt de la nouvelle DVF dans le cas de coupures de communication, nous utilisons deux environnements de simulation issus de Stage (Fig. 5).

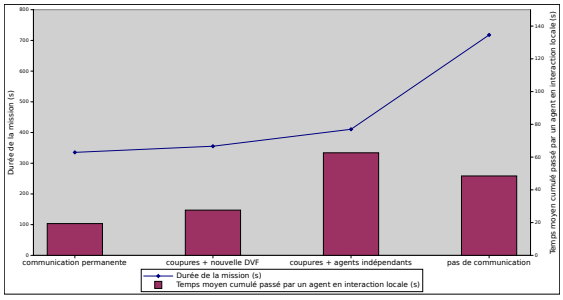


(a) office-like1 (b) office-like2

FIGURE 5 – Environnements de simulation et positions de départ.



(a) office-like1 avec 4 robots.



(b) office-like2 avec 3 robots.

FIGURE 6 – Résultats moyennés sur 5 simulations.

Nous avons choisi ces environnements car ils sont denses. Dans de tels environnements, de nombreuses situations d'interaction locale peuvent survenir et c'est la principale difficulté à surmonter dans des environnements denses. Or les interactions locales sont un bon indicateur de la qualité de la politique calculée avec la nouvelle DVF. Avec des environnements aléatoires, nous aurions obtenu des environnements plus clairsemés et donc moins de situations potentielles d'interaction locale. Les environnements choisis sont donc plus intéressants étant donné qu'ils vont mettre en évidence l'intérêt de la nouvelle DVF. Dans des environnements clairsemés, la nouvelle DVF serait aussi performante mais nous n'aurions pas l'indicateur d'interaction locale.

Schémas de communication et stratégies d'exploration Nous testons 4 situations avec des schémas de communication variés et différentes stratégies d'exploration : pas de communication *i.e.* que les robots sont indépendants et calculent des stratégies avec l'algorithme d'itération sur les valeurs standard (eq. 1) ; communication permanente et calcul des stratégies avec la nouvelle DVF (eq. 3) ; et coupures de communication. Dans ce dernier cas, les robots calculent la DVF quand la communication est disponible et pendant les coupures de communication, deux méthodes sont comparées. La première est que les robots sont indépendants pendant les coupures et chaque robot explore individuellement l'environnement (eq. 7.). Cette

méthode est utilisée dans la littérature par (Burgard *et al.*, 2005). La seconde est que les robots utilisent la nouvelle DVF (algorithme 1). A chaque simulation, 5 coupures sont générées de manière stochastique ; chacune dure 25 secondes dans l'environnement office-like1 et 40 secondes dans l'environnement office-like2.

Indicateur de la qualité de la politique calculée Nous traçons pour chaque situation la durée de la mission (exploration de l'environnement et retour à la zone de départ de tous les robots) et le temps moyen cumulé passé par tous les agents durant une mission en situations d'interaction locale. En effet, la sous-optimalité provient des situations d'interaction locale. Une politique jointe optimale doit minimiser ces interactions locales et c'est pour cela que nous utilisons le temps passé par les agents dans ces situations comme un indicateur de la qualité de la politique obtenue. Les situations d'interaction locale sont définies comme une distance entre deux robots inférieures à 1 mètre. Pendant ces situations, la coordination locale est requise si elle peut être appliquée, c'est-à-dire si les robots peuvent communiquer.

Résultats Les résultats sont présentés à la figure 6. Les agents indépendants (pas de communication) finissent la mission avec la durée la plus élevée car les robots explorent individuellement et chacun couvre tout l'espace. Avec une communication permanente, les missions d'exploration sont les plus rapides et les situations d'interaction locale sont minimisées car les robots se partagent efficacement la zone à explorer. Dans le cas de coupures de communication, la nouvelle DVF réduit radicalement les situations d'interaction locale comparée à l'approche existante dans la littérature (agents indépendants pendant les coupures). Le temps d'exploration est légèrement supérieur au temps mis par les robots dans le cas d'une communication permanente. Ainsi, cela montre que les robot utilisant la nouvelle DVF réussissent à se coordonner même s'il y a des coupures de communication. Les situations d'interaction locale, indicateur de la qualité de notre politique, montrent que notre approche permet de réduire la plupart de ces situations qui sont particulièrement risquées durant une coupure de communication où les collisions sont évitées uniquement par le système de bas-niveau d'anti-collision.

De plus, il est intéressant de remarquer que les interactions locales sont plus ou moins importantes sans communication selon la structure de l'environnement et en particulier, selon les situations des agents lorsque la communication coupe. En effet si une coupure intervient quand les agents sont éloignés, il ne rentreront pas en situation d'interaction locale. Dans certains environnements comme à la figure 5b, les situations d'interaction locale sont moins fréquentes même sans communication car la structure de l'environnement permet à des agents indépendants d'être rarement proches les uns des autres. Cependant la nouvelle DVF réduit toujours la coordination locale comparé aux autres approches.

7 Conclusion

Dans cet article, nous nous intéressons au problème d'exploration multi-robot sous contraintes de communication avec une résolution orientée interaction du modèle Dec-MDP. Nous étendons la technique par fonction de valeur distribuée qui suppose une communication gratuite et permanente et proposons une nouvelle DVF qui prend en compte des coupures potentielles de communication entre les robots. Nous appliquons cette méthode à des scénarios d'exploration multi-robot, afin que chaque robot calcule localement une stratégie qui minimise les interactions entre les robots et maximise la couverture de l'espace. Des résultats expérimentaux sont présentés, issus de scénarios avec des robots réels et simulés. Ces résultats ainsi que notre position de vice-champion à un défi robotique d'exploration multi-robot montre que notre méthode permet la coordination efficace d'une équipe de robots mobiles au cours de l'exploration et est robuste à des coupures de communication.

Références

BABVEY S., MOMTAHAN O. & MEYBODI M. R. (2003). Multi mobile robot navigation using distributed value function reinforcement learning. In *Proc. of ICRA*, p. 957–962.

BAUTIN A., SIMONIN O. & CHARPILLET F. (2011). Towards a communication free coordination for multi-robot exploration. In *6th National Conference on Control Architectures of Robots*.

BELLMAN R. (1957). *Dynamic programming : Markov decision process*.

BERNSTEIN D. S., GIVAN R., IMMERMAN N. & ZILBERSTEIN S. (2002). The complexity of decentralized control of markov decision processes. *Math. Oper. Res.*, **27**, 819–840.

BEYNIER A. & MOUADDIB A.-I. (2005). A polynomial algorithm for decentralized markov decision processes with temporal constraints. In *Proc. of AAMAS*, p. 963–969.

BEYNIER A. & MOUADDIB A.-I. (2006). An iterative algorithm for solving constrained decentralized markov decision processes. In *Proc. of AAAI*, p. 1089–1094.

BOUTILIER C. (1996). Planning, learning and coordination in multiagent decision processes. In *TARK*.

BURGARD W., MOORS M., STACHNISS C. & SCHNEIDER F. (2005). Coordinated multi-robot exploration. *IEEE Transactions on Robotics*, **21**, 376–386.

CANU A. & MOUADDIB A.-I. (2011). Collective decision- theoretic planning for planet exploration. In *Proc. of ICTAI*.

GLOANNEC S. L., JEANPIERRE L. & MOUADDIB A.-I. (2010). Unknown area exploration with an autonomous robot using markov decision processes. In *Proc. of Towards Autonomous RObotic Systems*, p. 119–125.

HOOG J., CAMERON S. & VISSER A. (2010). Autonomous multi-robot exploration in communication-limited environments. In *Towards Autonomous Robotic Systems*.

MATIGNON L., JEANPIERRE L. & MOUADDIB A.-I. (2012a). Distributed value functions for multi-robot exploration. In *Proc. of ICRA*.

MATIGNON L., JEANPIERRE L. & MOUADDIB A.-I. (2012b). Distributed value functions for the coordination of decentralized decision makers (short paper). In *Proc. of AAMAS*.

MELO F. S. & VELOSO M. M. (2011). Decentralized mdps with sparse interactions. *Artif. Intell.*, **175**(11), 1757–1789.

NAIR R., VARAKANTHAM P., TAMBE M. & YOKOO M. (2005). Networked distributed pomdps : A synthesis of distributed constraint optimization and pomdps. In *Proc. of AAAI*, p. 133–139.

POWERS M., BALCH T. & LAB B. (2004). Value-based communication preservation for mobile robots. In *Proc. of 7th International Symposium on Distributed Autonomous Robotic Systems*.

PUTERMAN M. L. (1994). *Markov decision processes*. John Wiley and Sons.

PYNADATH D. & TAMBE M. (2002). Multiagent teamwork : Analyzing the optimality and complexity of key theories and models. *Journal of Artificial Intelligence Research*, **16**, 389–423.

SCHNEIDER J., WONG W.-K., MOORE A. & RIEDMILLER M. (1999). Distributed value functions. In *Proc. of ICML*, p. 371–378.

SPAAN M. T. J. & MELO F. S. (2008). Interaction-driven markov games for decentralized multiagent planning under uncertainty. In *Proc. of AAMAS*, p. 525–532.

VELAGAPUDI P., VARAKANTHAM P., SYCARA K. & SCERRI P. (2011). Distributed model shaping for scaling to decentralized pomdps with hundreds of agents. In *Proc. of AAMAS*, p. 955–962.

WURM K. M., STACHNISS C. & BURGARD W. (2008). Coordinated multi-robot exploration using a segmentation of the environment. In *Proc. of IROS*, p. 1160–1165.

XIE J., NASHASHIBI F., PARENT N. M. & GARCIA-FAVROT O. (2010). A real-time robust slam for large-scale outdoor environments. In *17th ITS World Congress*.

ZLOT R., STENTZ A., DIAS M. & THAYER S. (2002). Multi-robot exploration controlled by a market economy. In *Proc. of ICRA*, volume 3, p. 3016–3023.