

Stratégie d'exploration multirobot fondée sur les champs de potentiels artificiels

Antoine Bautin, Olivier Simonin, François Charpillet

► To cite this version:

Antoine Bautin, Olivier Simonin, François Charpillet. Stratégie d'exploration multirobot fondée sur les champs de potentiels artificiels. Revue des Sciences et Technologies de l'Information - Série RIA : Revue d'Intelligence Artificielle, Lavoisier, 2012, Systèmes multi-agents - Initiatives mixtes, 26 (5), pp.523-542. 10.3166/ria.26.523-542 . hal-00757894

HAL Id: hal-00757894

<https://hal.inria.fr/hal-00757894>

Submitted on 27 Nov 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Stratégie d'exploration multirobot fondée sur les champs de potentiels artificiels

Antoine Bautin, Olivier Simonin, François Charpillet

Équipe MAIA

Université de Lorraine, LORIA, UMR 7503, Vandœuvre-lès-Nancy, F-54500, France

Inria, Villers-lès-Nancy, F-54600, France

prenom.nom@loria.fr

RÉSUMÉ. Le contexte de ce travail est le problème de l'exploration multirobot d'environnements inconnus pour la construction d'une carte métrique. Nous proposons une nouvelle technique de déploiement distribuée, basée sur le choix de la frontière à explorer (limites entre les zones accessibles et inexplorées) en fonction de la position (rang relatif en termes de distance) du robot par rapport à ses voisins. Un champ de potentiels sans minimum local est calculé depuis chaque frontière permettant de connaître en tout point le chemin le plus court pour s'y rendre. L'exploration consiste à affecter régulièrement à chaque robot une frontière à atteindre. L'assignation d'une frontière à un robot est réalisée par un nouvel algorithme utilisant comme critère le nombre de robots plus proches de la frontière considérée. Des mesures de performance en simulation et avec des robots sont présentées et montrent la validité et l'efficacité de notre approche en comparaison à d'autres approches.

ABSTRACT. The context of this work is multirobot exploration and mapping of unknown environments. Frontiers (boundaries between empty and unexplored areas) are points to go to in order to increase explored area. In this paper, we propose a frontier allocation method for the exploration and mapping of indoor environments using multiple robots. A wavefront propagation algorithm is used to build a local minimum free potential field with a gradient descending towards frontiers used for navigation and path distance computation. Exploration is done by successively exploring frontiers assigned using a novel, computationally inexpensive algorithm based on the cardinal of the set of robots closer to the frontier than the robot being assigned. Experiments in simulation and on robots demonstrated the validity of the frontier allocation approach.

MOTS-CLÉS : exploration multirobot, allocation de tâches décentralisée, navigation par champs de potentiels artificiels

KEYWORDS: multirobot exploration, decentralized multirobot task allocation, artificial potential field navigation method

DOI:10.3166/RIA.26.523-542 © 2012 Lavoisier

1. Introduction

L'exploration d'un environnement inconnu est un problème fondamental en robotique mobile. Elle permet en particulier de construire une carte utile dans de nombreuses applications (reconnaissance, sécurité, assistance, etc). La création d'une carte nécessite la résolution de trois sous-problèmes, la localisation du ou des robots, la cartographie des zones explorées et l'exploration, au sens de la découverte d'un espace inconnu. Nous nous intéressons dans cet article à ce dernier problème.

Nous cherchons une stratégie permettant aux robots de visiter toutes les zones de l'environnement le plus rapidement possible. Dans ce contexte, l'utilisation de plusieurs robots est avantageuse mais le gain est conditionné par le niveau de coopération entre les robots.

Une stratégie de déploiement vise à minimiser le temps d'exploration total (temps nécessaire à la construction de la totalité de la carte) en répartissant les robots dans l'environnement pour permettre l'exploration simultanée de zones distinctes de l'environnement. Nous considérons dans cette étude que la flottille est homogène (les robots la composant sont identiques), que les robots peuvent, indépendamment, se localiser et construire une carte à partir de leur perception locale, et qu'ils peuvent communiquer et fusionner leurs cartes respectives. Les principaux problèmes restant à résoudre sont le choix des sous-tâches que chaque robot devra accomplir, la méthode d'affectation des sous-tâches aux robots et la fréquence à laquelle leurs affectations sont recalculées.

La principale contribution de cet article est un nouvel algorithme pour l'affectation de frontières aux robots. Chaque robot est affecté à la frontière pour laquelle il est en meilleure position, c'est-à-dire la frontière où il y a le moins de robots plus proches que lui. Cette proximité est définie comme la distance du chemin à parcourir jusqu'à la frontière qui est calculée à l'aide d'un champ de potentiels. Des expériences en simulation montrent la validité et l'efficacité de l'approche proposée tout en ayant une complexité inférieure aux algorithmes classiques.

La section 2 pose plus précisément le problème traité ainsi que les critères d'évaluation d'une assignation. Dans la section 3, les principales approches existantes pour l'exploration multirobot sont exposées. La section 4 décrit l'approche proposée, qui est ensuite analysée et comparée expérimentalement aux approches existantes dans la section 5. Enfin, nous concluons et dégageons des perspectives à ce travail.

2. Spécification du problème

L'environnement à explorer est un espace fini dans lequel est déployé un ensemble de robots mobiles autonomes (les agents) dotés de capacités d'observation leur permettant d'établir une carte locale de l'environnement, de se situer dans la carte construite. En pratique, ceci est possible en équipant les robots d'un télémètre et en utilisant un algorithme de localisation et cartographie simultanée (*Simultaneous Localization and Mapping*) comme, par exemple, celle introduite par Thrun et Liu (2005)

ou Howard (2006)). La communication est essentielle pour la coopération, nous considérons donc que les robots peuvent communiquer entre eux. Nous utilisons une représentation de cartes métriques de type grille d'occupation (Elfes, 1989) sur laquelle des cellules frontières à explorer sont identifiées. Le problème que nous traitons consiste alors à assigner à chaque robot une frontière à explorer, c'est-à-dire à s'en rapprocher pour étendre la connaissance de l'environnement. L'évolution de la carte implique une réassignation continue des agents vers les frontières nouvellement identifiées. Il s'agit d'un problème dynamique. Toutefois il est généralement traité comme un problème d'assignation robots-frontières statique effectué régulièrement.

Le problème d'allocation des frontières aux robots est un problème d'optimisation combinatoire NP-difficile. Plus précisément, on cherche une assignation parmi tous les arrangements sans répétition soit $\frac{m!}{(m-n)!}$ assignations possibles quand $n \leq m$ où $n = |\mathcal{R}|$ est le nombre d'agents et $m = |\mathcal{F}|$ le nombre de frontières. Cette quantité rend la recherche d'optimalité rapidement intraitable quand le nombre de frontières et de robots croît, une approximation est donc nécessaire.

Pour évaluer ces assignations il faut être capable de connaître le coût de parcours des robots vers les frontières. Dans les approches présentées ci-après, ce coût est généralement calculé avec un algorithme A*. Cette opération fournissant une matrice de coûts associant frontières et robots.

Avant de présenter des approches existantes, nous précisons les notations employées ainsi que les critères de qualité d'assignation.

2.1. Notations

- soit \mathcal{E} l'environnement, \mathcal{E}_{exp} la partie de l'environnement déjà explorée et \mathcal{E}_{inexp} la partie de l'environnement inexploré
- soit \mathcal{R} l'ensemble des robots, $\mathcal{R}_1 \dots \mathcal{R}_n$ avec n le nombre de robots $|\mathcal{R}|$
- soit \mathcal{F} l'ensemble des frontières, $\mathcal{F}_1 \dots \mathcal{F}_m$ avec m le nombre de frontières $|\mathcal{F}|$
- soit \mathcal{C}_{ij} le coût associé à l'assignation du robot \mathcal{R}_i à la frontière \mathcal{F}_j
- soit \mathcal{A} une matrice d'assignations composée de $\alpha_{ij} \in [0, 1]$ tel que :

$$\alpha_{ij} = \begin{cases} 1 & \text{si le robot } \mathcal{R}_i \text{ est assigné à } \mathcal{F}_j \\ 0 & \text{sinon} \end{cases}$$

2.2. Critères d'évaluation d'une assignation

Pour déployer les robots dans l'environnement nous affectons à chaque robot une frontière à explorer. Cette contrainte s'exprime par :

$$\forall i \quad \sum_{j=1}^m \alpha_{ij} = 1$$

L'opération d'affectation est effectuée à chaque changement d'état du système (exploration, disparition ou création d'une frontière). La qualité d'une assignation est évaluée par le respect de trois critères détaillés ci-après.

1. Équilibre de la répartition des robots sur les frontières. Lors de l'exploration nous rencontrons trois cas de figures :

– *Nombre de robots égal au nombre de frontières.* Cas le plus simple, une frontière devra être affectée à chaque robot.

– *Nombre de frontières supérieur au nombre de robots.* Les robots devront chacun être affectés à des frontières distinctes pour répartir les robots vers des zones inexplorées différentes et ainsi éviter la redondance des informations récoltées.

– *Nombre de robots supérieur au nombre de frontières.* Après l'affectation d'un robot à chaque frontière, il faut déterminer une stratégie pour les robots supplémentaires. Au regard du critère de minimisation du temps total d'exploration, laisser des robots immobiles n'est pas une bonne stratégie : si un robot découvre une vaste zone inexplorée de l'environnement plusieurs robots seront nécessaires pour l'explorer plus rapidement. Comme il n'y a pas d'information, a priori, sur ce qui se trouve derrière les frontières, une répartition équilibrée du nombre de robots par frontière est souhaitable.

Ce critère d'affectation équilibré peut se résumer par :

$$\lfloor n/m \rfloor \leq \forall j \sum_{i=1}^n \alpha_{ij} \leq \lceil n/m \rceil \quad (1)$$

le nombre de robots par frontière est égal à une différence d'un robot près.

2. Minimum de la somme des coûts Il s'agit de minimiser le coût de l'exploration en minimisant la somme des coûts (le coût de parcours de chaque robot) :

$$C(\mathcal{A}) = \sum_{i=1}^n \sum_{j=1}^m \alpha_{ij} C_{ij} \quad (2)$$

3. Minimum du maximum des coûts Le respect du critère 2 ne garantit pas une solution unique. Par exemple, la figure 1 illustre une situation où 2 solutions existent respectant les critères 1 et 2 mais celle affichée à droite s'effectuera en un temps d'exploration inférieur. En effet le temps d'exploration de toutes les frontières affectées aux robots est déterminé par le temps d'exploration individuel maximum. Un troisième critère d'évaluation est donc la minimisation du coût maximum :

$$C_{max}(\mathcal{A}) = \max_{\forall i} \sum_{j=1}^m \alpha_{ij} C_{ij} \quad (3)$$

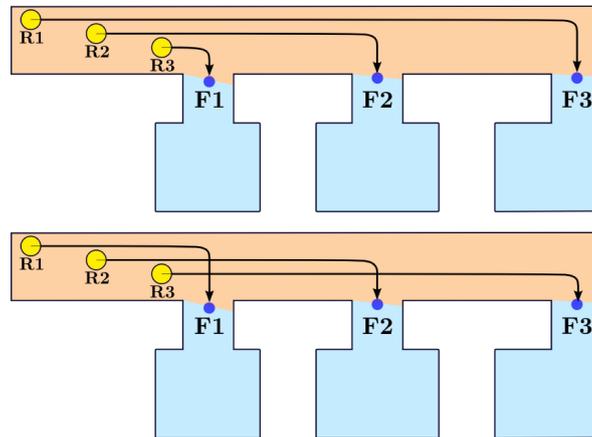


Figure 1. Illustration du critère 3 : la somme des coûts de chaque assignation est identique mais le temps pour réaliser l'exploration de chaque frontière sera inférieur dans l'assignation du bas

2.3. Calcul centralisé versus décentralisé

Un critère à prendre en compte dans le choix de la méthode d'assignation est le coût de communication. En effet selon que le calcul sera centralisé ou décentralisé la méthode n'aura pas le même coût ni la même robustesse. Sous l'hypothèse de la disponibilité de la matrice de coûts, l'assignation de frontières peut être effectuée soit par un agent central (robot dans l'environnement ou serveur central) soit en décentralisant la décision à travers des agents communicants. La dernière possibilité est de laisser les agents décider de manière autonome, c'est-à-dire d'avoir un système décentralisé sans communication.

3. État de l'art

Dans cette section nous présentons les principaux types d'approches existantes pour l'assignation de frontières aux robots pour l'exploration d'environnements inconnus.

Les premières méthodes d'exploration étaient basées sur une navigation aléatoire (Mantaras *et al.*, 1997) ou le suivi de mur (Duckett, Nehmzow, 1997). En monorobot, comme en multirobot, une technique admise comme efficace est l'exploration successive des frontières créées entre les zones non explorées et les zones explorées accessibles (Yamauchi, 1997). Les robots se déplacent vers les frontières et collectent ainsi de nouvelles informations sur les zones inexplorées de l'environnement.

3.1. Frontière la plus proche

Yamauchi est le premier à avoir développé une stratégie multirobot d'exploration de frontière (Yamauchi, 1998). Les robots partagent les informations locales récoltées afin que chacun d'entre eux produise une carte similaire fournissant une liste de frontières similaires. Chaque robot se dirige vers la frontière la plus proche de lui, effectue une observation et en diffuse (*broadcast*) le résultat. Aucune communication n'est nécessaire pour coordonner les robots. C'est un système asynchrone, distribué et robuste aux pannes de robots. L'algorithme 1 explicite le calcul effectué par chaque robot pour choisir sa frontière. Lors de son application, une coopération implicite s'effectue (voir la figure 2) mais elle est limitée car les robots peuvent choisir les mêmes frontières ne tirant pas avantage de leur nombre. La figure 3 illustre ce type de situation. Notons que l'algorithme 1 est optimal au regard du critère (2) de coût d'exploration à chaque étape d'assignation mais ne respecte pas le critère (1) d'équilibre de la répartition des robots sur les frontières.

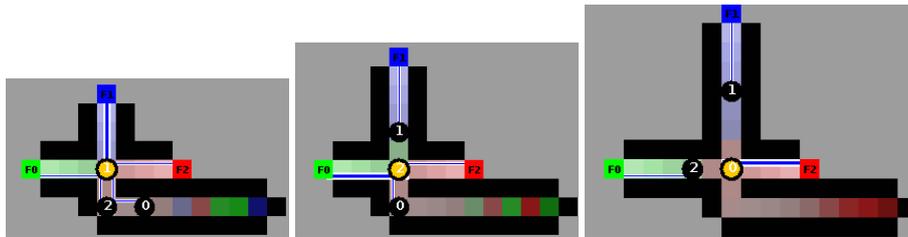


Figure 2. Coordination implicite avec l'algorithme Frontière la plus proche. Description de gauche à droite : quand le premier robot \mathcal{R}_1 atteint l'intersection des couloirs il découvre 3 frontières équidistantes et choisit aléatoirement la frontière \mathcal{F}_1 , quand \mathcal{R}_2 atteint l'intersection, il reste 2 frontières plus proches équidistantes et choisit aléatoirement \mathcal{F}_0 entre \mathcal{F}_0 et \mathcal{F}_2 , quand le 3^{ième} robot \mathcal{R}_0 arrive sur l'intersection une seule frontière est plus proche de lui et il choisit donc \mathcal{F}_2

Algorithme 1 : Frontière la plus proche

Entrées : \mathcal{C}_i un vecteur de coût de chemin à chaque frontière

Sorties : \mathcal{A}_i l'assignation du robot i

début

| $\alpha_{ij} = 1$ tel que $j = \underset{\forall \mathcal{F}_j \in \mathcal{F}}{\operatorname{argmin}} \mathcal{C}_{ij}$

fin

La complexité computationnelle de cet algorithme est $O(m)$.

3.2. Glouton

L'algorithme *Glouton* tente d'optimiser le coût total de l'exploration (critère (2)) en fixant à chaque itération la paire robot-frontière ayant le coût minimum. Comme il

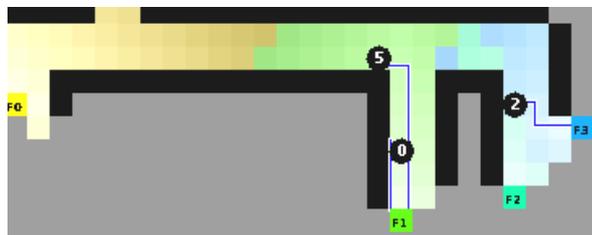


Figure 3. Résultat de l'assignation avec l'algorithme Frontière la plus proche.
Les robots ne prennent pas en compte la position des autres robots.
Les robots \mathcal{R}_5 et \mathcal{R}_0 sont assignés à la même frontière

itére sur les robots, le respect du critère (1) d'équilibre est garanti. Cet algorithme est en général appliqué de manière centralisée mais si chaque robot dispose de la matrice de coûts, son exécution décentralisée est possible (chaque agent exécute l'algorithme jusqu'à obtenir son assignation), cf. algorithme 2. La figure 4 illustre l'exécution de cet algorithme. Comme nous le détaillons ci-après, la plupart des approches d'allocation de frontières sont basées sur cet algorithme (Burgard *et al.*, 2005) (Simmons *et al.*, 2000) (Zlot *et al.*, 2002).

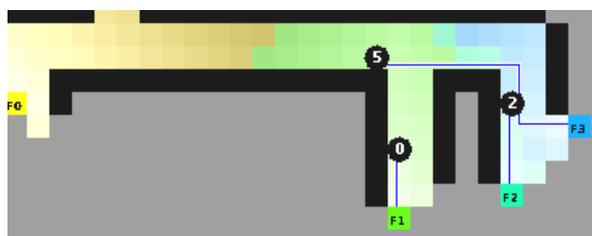


Figure 4. Illustration de l'allocation de frontière avec l'algorithme Glouton

Algorithme 2 : *Glouton* (version décentralisée)

Entrées : \mathcal{C} Matrice de coûts

Sorties : α_{ij} assignation du robot \mathcal{R}_i à la frontière \mathcal{F}_j

tant que \mathcal{R}_i n'est pas assigné **faire**

Trouver $i, j = \operatorname{argmin} C_{ij} \forall \mathcal{R}_i \in \mathcal{R}, \forall \mathcal{F}_j \in \mathcal{F}$

$\alpha_{ij} = 1$

$\mathcal{R} = \mathcal{R} \setminus \mathcal{R}_i$

$\mathcal{F} = \mathcal{F} \setminus \mathcal{F}_j$

SI $\mathcal{F} = \emptyset$ ALORS $\mathcal{F} = \mathcal{F}_{init}$

fin

La complexité de l'algorithme 2 (*Glouton*) est $O(n^2m)$.

3.3. *Glouton sur l'utilité des frontières*

Burgard *et al.* (2005) utilisent l'algorithme *Glouton* en intégrant un critère d'utilité des frontières, initialement proposé par Simmons *et al.* (2000). Il est défini comme la différence ou le rapport entre le coût et le gain d'information lié à l'observation de la frontière. Le gain d'information est une estimation de la taille de la zone qui sera découverte. L'utilité vise à réduire l'attractivité des frontières proches de celles déjà assignées. Ainsi, lorsqu'une frontière est affectée à un robot, les gains d'information associés à cette frontière et aux frontières visibles depuis celle-ci sont réduits de façon inversement proportionnelle à leur interdistance. Cela permet d'obtenir une meilleure répartition des robots dans l'environnement si les frontières sont assez proches pour qu'un robot les explore lors d'une même observation. L'algorithme est semblable au *Glouton* avec la mise à jour du critère d'importance après chaque assignation et sa complexité computationnelle est du même ordre $O(n^2m)$. Les cellules frontières voisines peuvent aussi être regroupées en une frontière unique ou l'identification des frontières peut se faire en utilisant la méthode proposée par Franchi *et al.* (2009). Ces approches, comme celles fondées sur l'utilité visent à éviter l'assignation de plusieurs robots à des cellules frontières proches l'une de l'autre.

3.4. *Système d'enchères*

Les systèmes d'enchères fonctionnent en deux temps, tout d'abord une phase d'enchères basée sur des communications entre robots ou bien avec un serveur central, puis une phase d'assignation. L'assignation de frontières utilise généralement le même principe que l'algorithme *Glouton*. Pratiquement, chaque robot émet une enchère sur chaque frontière, il remporte une frontière pour laquelle il a la meilleure enchère.

Les frontières peuvent être mises aux enchères simultanément, ainsi, Simmons *et al.* proposent une approche basée sur l'exploration de frontière et un système d'enchère simple (Simmons *et al.*, 2000). Chaque robot évalue le profit de l'exploration de chaque frontière en déterminant son coût de déplacement et une estimation du futur apport d'information. Il envoie ensuite une enchère sur les frontières évaluées à un agent central qui, une fois reçues les enchères de tous les robots, assigne les frontières aux robots de manière gloutonne. Zlot *et al.* ont développé une méthode distribuée similaire où les robots qui découvrent une frontière sont commissaires-priseurs pour la frontière découverte, les autres robots sont enchérisseurs (Zlot *et al.*, 2002).

3.5. *Conclusion de l'état de l'art*

Les approches efficaces connues reposent sur l'algorithme *Glouton* qui a une complexité élevée $O(n^2m)$. Seul l'algorithme *Frontière la plus proche* a une faible complexité mais n'assure pas une répartition équilibrée (critère (1)). Le modèle que nous proposons ci-après est aussi efficace que le *Glouton* tout en ayant une complexité inférieure.

4. Approche proposée

4.1. Principe

Les approches existantes répartissent les robots sur les frontières en fonction des distances. Ce type d'affectation peut assigner plusieurs robots proches à des frontières situées dans une même "direction". Nous qualifions de même direction des frontières situées dans une zone qui n'est atteignable par ces robots qu'en passant par un même chemin. Par conséquent, il y a un risque de redondance pour l'exploration d'une direction donnée, alors que des frontières plus éloignées peuvent rester orphelines (sans affectation). Cette situation est illustrée en figure 4, où les agents 2 et 5 se dirigent vers la même direction tandis que la frontière \mathcal{F}_0 est orpheline. Pour résoudre ce problème, **nous calculons pour chaque couple robot-frontière le nombre de robots plus proches de la frontière considérée**. Ce nombre est appelé *position du robot sur la frontière*. Chaque robot est affecté à la frontière pour laquelle il a la meilleure position. Notre proposition est donc une alternative aux approches centrées distances aux frontières par l'utilisation de la position vers les frontières.

Pour évaluer ces positions nous utilisons la matrice de coûts. Pour la calculer, nous construisons un champ de potentiel sans minimum local depuis chaque frontière (algorithme de front de vague (Barraquand *et al.*, 1992)). Il fournit la distance du chemin à parcourir pour atteindre la frontière en tout point de l'environnement et donc pour chaque robot. De plus, la descente du gradient permet de naviguer jusqu'à la frontière. En comparaison, les approches vues dans l'état de l'art calculent un chemin du robot à chaque frontière. Si un robot veut connaître le coût pour un autre robot il devra soit le lui demander soit calculer le chemin qu'il emprunterait.

Le principe général de l'approche présentée ci-après, sous l'hypothèse d'une carte partagée, peut être résumé par les différentes étapes suivantes :

1. Identification des frontières entre régions explorées/inexplorées
2. Calcul des champs de potentiels depuis les frontières
3. Attribution des frontières aux robots

Chaque robot se dirige alors vers la frontière qui lui est attribuée. Ces étapes peuvent être effectuées de façon décentralisée (sur chaque robot) ou sur un serveur central. Nous détaillons maintenant les étapes 2 et 3.

4.2. Propagation du front de vague

Nous supposons que l'environnement est discrétisé en cellules de taille équivalente à un robot et contenant une information sur son statut (*occupé*¹, *vide* ou *inexploré*). L'algorithme 3 précise le calcul de la vague. Il consiste à calculer incrémentalement

1. Par occupé nous entendons occupé par un obstacle et non par un robot.

l'ensemble des cellules situées à la même distance de la source² – le front de vague –, à partir de la source initialisée à 0. Cet algorithme a une complexité en $O(n)$ où n est le nombre de cellules atteignables. Notons que pour le calcul des positions, l'exécution de l'algorithme de propagation de front de vague peut être arrêté dès que le front de vague atteint la cellule occupée par le robot qui calcule son assignation, réduisant ainsi le coût du calcul du champs de potentiels. En effet, lorsque le robot est atteint, tous les robots plus proches l'on déjà été.

Soit \mathcal{W}_i , l'ensemble des cellules du front de vague à la distance i , $\phi : \mathcal{X} \rightarrow [0, \infty[$ la fonction de potentiel avec \mathcal{X} les cellules de l'environnement. Notons \mathcal{X}_G la cellule but (dans notre cas la frontière) de potentiel 0.

Algorithme 3 : Algorithme de propagation de front vague

- 1 Initialiser $\mathcal{W}_0 = \mathcal{X}_G; i = 0$
 - 2 Initialiser $\mathcal{W}_{i+1} = \emptyset$
 - 3 Pour chaque $x \in \mathcal{W}_i$, $\phi(x) = i$ et insérer toutes les cellules voisines de x dans \mathcal{W}_{i+1}
 - 4 SI $\mathcal{W}_{i+1} = \emptyset$ ALORS terminer SINON $i=i+1$ et aller à l'étape 2.
-

La figure 5 illustre l'exécution de l'algorithme 3. Dans les figures 2, 3, 4 et 6 les cellules sont colorées par le gradient provenant de la frontière la plus proche.

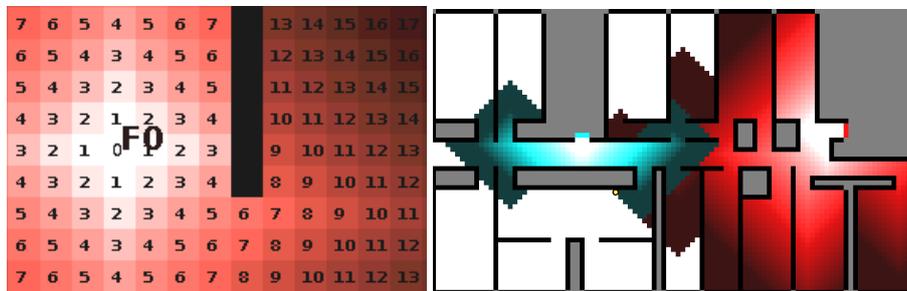


Figure 5. Résultat du champ de potentiels calculé avec l'algorithme 3; à gauche : une cellule frontière et les valeurs affichées sont les distances à la frontière ; à droite deux frontières, la couleur d'une cellule est la couleur associée à la frontière la plus proche et sa luminosité décroît en fonction de la distance à la frontière

4.3. Affectation des frontières aux robots

L'algorithme que nous proposons cherche à distribuer les robots vers les différentes frontières en minimisant le nombre de robots empruntant les mêmes directions

2. Distance du chemin parcouru depuis la source.

(mêmes espaces traversés par des robots). Pour cela nous posons la notion de position d'un robot vers une frontière comme étant le nombre de robots plus proches de la frontière que lui. Formellement, \mathcal{P}_{ij} la position du robot i vers la frontière j est définie par :

$$\mathcal{P}_{ij} = \sum_{\forall k \in \mathcal{R}_k, k \neq i, C_{kj} < C_{ij}} 1$$

Nous en déduisons l'algorithme 4 nommé *Position Minimum*, permettant à un robot de trouver la frontière à laquelle il est assigné. Un robot est assigné à la frontière pour laquelle il est en meilleure position (la plus petite). En cas d'égalité entre des frontières, le robot choisit la plus proche.

En raisonnant sur les positions plutôt que sur les distances, 2 robots voisins se répartiront sur 2 frontières dans des directions distinctes où ils seront premiers quelles que soient les distances. Nous verrons dans la section qui suit que cette répartition dans des directions distinctes favorise le critère 1 d'équilibre de la répartition.

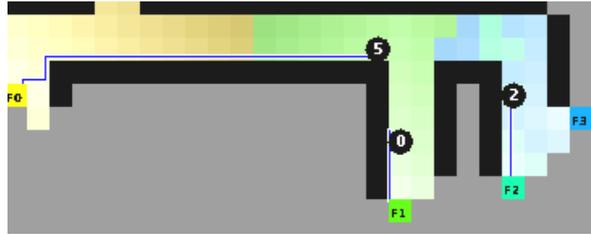


Figure 6. Résultat de l'assignation avec l'algorithme 4 *Position Minimum*

Algorithme 4 : *Position Minimum*

Entrées : \mathcal{C} matrice de coûts

Sorties : α_{ij} assignation du robot \mathcal{R}_i

1 **pour chaque** $\mathcal{F}_j \in \mathcal{F}$ **faire**

2 $\mathcal{P}_{ij} = \sum_{\forall k \in \mathcal{R}_k, k \neq i, C_{kj} < C_{ij}} 1$

3 **fin**

4 $\alpha_{ij} = 1$ tel que $j = \operatorname{argmin}_{\forall \mathcal{F}_j \in \mathcal{F}} \mathcal{P}_{ij}$

5 *En cas d'égalité choisir la frontière avec le coût minimum parmi les minima de \mathcal{P}_{ij}*

La complexité de l'algorithme 4 *Position Minimum* est en $O(nm)$.

5. Résultats et analyse

Cette section analyse et compare les performances de l'algorithme proposé.

5.1. Comparaison aux méthodes existantes

L'évaluation de la méthode proposée a été effectuée à l'aide de simulations. Le modèle utilisé est simple, l'environnement et le temps sont discrets, un agent occupe une cellule, connaît sa position et perçoit parfaitement à 360° autour de lui avec un rayon paramétrable. Les expériences ont été réalisées sur des environnements différents comme des labyrinthes générés aléatoirement, des grilles régulières ou de type bâtiment comme la section d'hôpital et écuries (voir figure 7). Les expériences se terminent lorsque les agents ont construit la totalité de la carte.

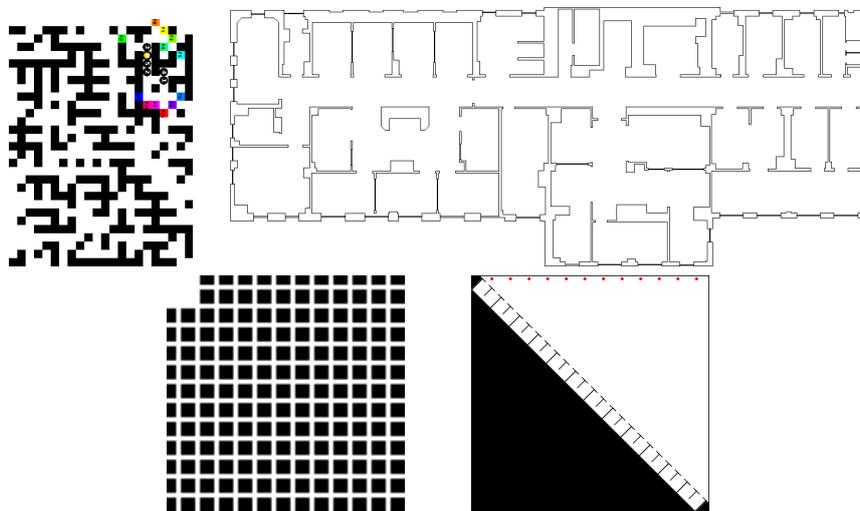


Figure 7. Environnements utilisés pour les expériences : labyrinthe, section d'hôpital de Player/Stage, grille régulière et écuries

Les figures 8 et 9 comparent les méthodes en temps d'exploration en faisant varier le nombre de robots. Les temps d'exploration sont donnés en pas de simulation. Les algorithmes comparés sont *Frontière la plus proche* (algorithme 1), une implémentation d'un algorithme *Glouton* (algorithme 2) fondée sur l'utilité : Burgard *et al.* (2005) et *Position Minimum* (algorithme 4).

Sur tous les environnements, nous observons que les algorithmes *Glouton* et *Position Minimum* sont nettement plus performants que l'algorithme *Frontière la plus proche*, l'écart est d'autant plus marqué que le nombre d'agents augmente (jusqu'à 20 %). Sur l'environnement labyrinthe (taille 60x60 cellules), les performances de *Glouton* et *Position Minimum* sont similaires avec un léger avantage pour *Position Minimum* lorsque le nombre de robots est faible. Une différence plus marquée apparaît sur l'environnement section d'hôpital : quand le nombre de robots est faible l'algorithme *Position Minimum* a de meilleures performances, en revanche, quand le nombre de robots devient plus important, l'approche de Burgard *et al.* a des performances légèrement meilleures. Plus généralement, nous avons observé que notre approche s'avère plus performante lorsque le nombre d'agents est faible par rapport au nombre de fron-

tières à explorer. En effet, l'approche gloutonne optimise seulement les distances et ne distribue pas les agents dans des directions différentes. Cela est abordé dans l'étude qualitative ci-après.

Sur l'environnement écuries, *Position Minimum* reste meilleur, de 18 % en moyenne, peu importe le nombre de robots. Ceci est illustré par le graphique 9. Cette différence s'explique par le fait que selon la proximité des frontières entre elles vis-à-vis de la position des robots, l'algorithme *Position Minimum* va empêcher des assignations à des frontières proches. C'est le cas avec cet environnement, où de nombreuses portes des pièces sont proches. La figure 10 illustre les vagues calculées depuis ces frontières, en les affichant lorsqu'elles sont en première position. L'assignation est ici optimale. En revanche, l'assignation gloutonne aurait alloué les frontières des trois premières demi-pièces aux trois robots dans leur ordre respectif.

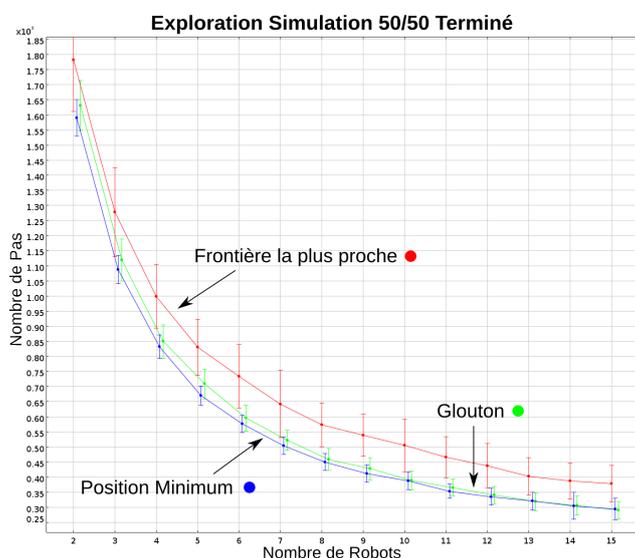


Figure 8. Temps d'exploration d'un labyrinthe 60x60 cellules, pour un nombre croissant d'agents (chaque valeur est une moyenne de 50 simulations)

Le tableau 1 montre la moyenne sur l'ensemble des environnements en nombre de pas de simulation ainsi que l'écart type associé. Il apparaît que *Position Minimum* a moins de variabilité. Ceci a été confirmé par un test d'analyse de variance.

Tableau 1. Résultats en simulation sur différents environnements

	Nb de pas moyen	Écart type
<i>Frontière la plus proche</i>	934.47	124.46
<i>Glouton</i>	717.66	42.76
<i>Position Minimum</i>	708.37	36.01

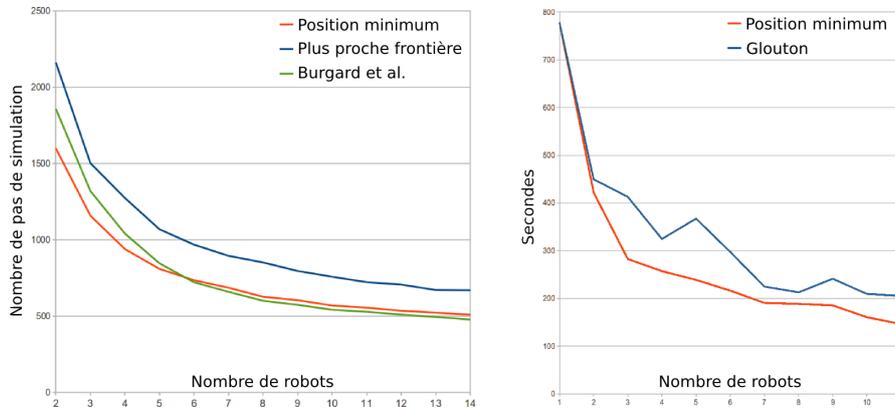


Figure 9. Temps d'exploration sur l'environnement section d'hôpital (chaque valeur est une moyenne de 60 simulations) (à gauche). Résultat de la simulation de l'exploration de l'environnement écuries (à droite)

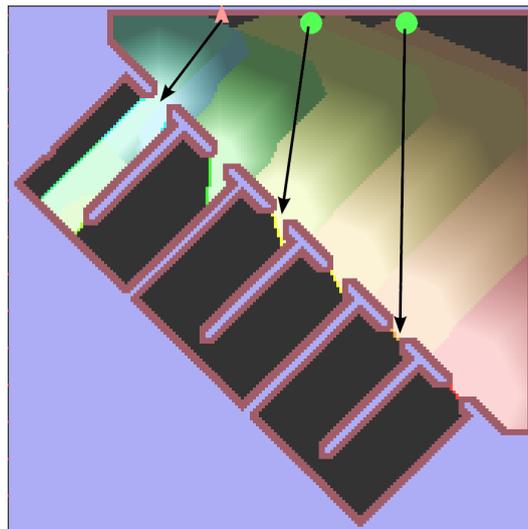


Figure 10. Illustration du calcul des vagues sur l'environnement écuries

5.2. Étude qualitative

Pour expliquer les résultats obtenus, nous analysons et comparons les affectations des différents algorithmes sur des exemples d'intersections.

En simulation, nous avons observé une meilleure répartition des agents avec *Position Minimum* qu'avec l'algorithme de *Frontière la plus proche* (Algo. 1). Une première analyse concerne la répartition en groupes de robots de tailles égales lorsque le nombre de frontières est inférieur à celui des robots. Cette situation est illustrée en

figure 11 sur un problème typique qui survient avec l'allocation par *Frontière la plus proche*. Un groupe de robots est proche d'une frontière, alors qu'une autre frontière se trouve plus loin dans une direction opposée. Tous les robots seront assignés à la même frontière tandis qu'avec l'algorithme *Position Minimum* ils seront séparés en deux groupes de tailles égales.

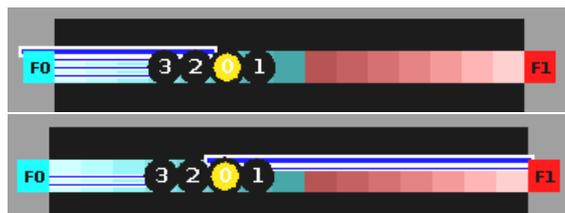


Figure 11. Différence entre l'allocation de frontières par l'algorithme *Frontière la plus proche* et *Position Minimum*. En haut : *Frontière la plus proche* : tous les robots sont assignés à la même frontière ; en bas : *Position Minimum* : les robots sont également répartis entre les 2 frontières

Par ailleurs, *Position Minimum* présente un comportement dynamique pertinent qui tend à séparer les robots groupés. Par exemple, dans la situation illustrée en figure 12, deux robots se suivent pour atteindre une même frontière. Ils se séparent dès que le premier s'éloigne de la frontière orpheline, celle-ci captant le second robot. En effet, il est alors en première position vers cette frontière. En comparaison, avec l'algorithme *Frontière la plus proche*, les deux robots auraient exploré la zone derrière la frontière \mathcal{F}_1 tandis qu'avec l'affectation gloutonne, les robots auraient initialement été affectés aux différentes frontières.

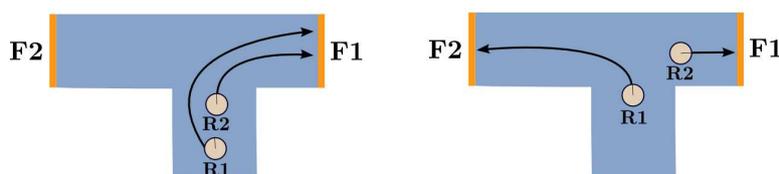


Figure 12. Deux robots se séparant sur une intersection en T avec l'algorithme *Position Minimum*

En comparaison avec l'affectation gloutonne, *Position Minimum* ne prend pas en compte uniquement la distance des frontières mais aussi la direction des frontières. Cette caractéristique a l'avantage de séparer les robots vers des directions différentes. Par exemple, dans la situation illustrée par la figure 13, l'affectation avec *Position Minimum* dirige le robot \mathcal{R}_1 vers la frontière \mathcal{F}_1 qui est plus éloignée que \mathcal{F}_2 et \mathcal{F}_3 mais qui est dans une direction différente. L'affectation gloutonne des frontières attribue \mathcal{F}_2 à \mathcal{R}_3 et \mathcal{F}_3 à \mathcal{R}_1 . Dans cette situation, l'exploration de toutes les frontières prendra moins de temps avec *Position Minimum* car les frontières \mathcal{F}_2 - \mathcal{F}_3 sont plus proches entre elles et pourront être explorées par \mathcal{R}_3 le temps que \mathcal{R}_1 navigue vers \mathcal{F}_1 .

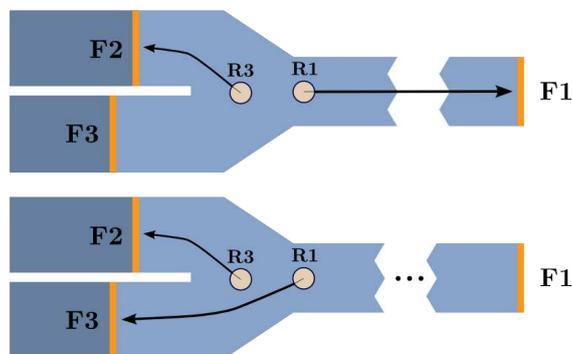


Figure 13. Comparaison des assignations avec les algorithmes *Position Minimum* (en haut) et *Glouton* (en bas)

Un avantage de l’algorithme *Glouton* par rapport à *Position Minimum* apparaît lorsque, du point de vue d’un robot, il existe une direction avec plusieurs frontières et plusieurs robots. *Position Minimum* prend en compte l’ensemble des robots dans une direction mais pas le nombre de frontières. Ainsi, si ce nombre de frontières est supérieur à celui des robots, il peut laisser des frontières sans robots alloués, alors qu’une répartition équilibrée était possible. La figure 14 illustre cette situation où le robot \mathcal{R}_3 “voit” la frontière \mathcal{F}_1 avec deux robots plus proches. Il est ainsi en troisième position pour les frontières \mathcal{F}_1 , \mathcal{F}_2 et \mathcal{F}_3 et va donc se diriger vers la frontière \mathcal{F}_4 pour laquelle il est en deuxième position. Dans cette situation, l’allocation gloutonne des frontières est plus efficace. Toutefois les résultats montrent que l’impact sur les performances sont faibles.

5.3. Expérimentation avec des robots

Dans cette section nous présentons le déploiement de l’algorithme *Position Minimum* avec des robots mobiles et établissons une première évaluation en situation réelle. Nous utilisons des robots “MiniRex” construits au laboratoire LISA de l’université d’Angers. Les robots sont équipés d’un télémètre laser Hokuyo utm-30lx d’une portée de 30 m et de trois capteurs ultrason. L’environnement à explorer est une arène d’approximativement 35 m² illustrée par la figure 15. Les robots démarrent d’une zone de départ connue qui leur permet de travailler dans le même repère orthonormé. Des vidéos des explorations réalisées en simulation et dans l’arène sont visibles à l’adresse suivante : www.loria.fr/~bautin.

La figure 15 montre la carte construite lors d’une exploration avec trois robots utilisant l’algorithme *Position Minimum* pour l’allocation des frontières. La grille d’occupation utilisée à une résolution de 2 cm² et celle où les frontières sont identifiées à une résolution de 8 cm². Les cellules frontières contiguës sont regroupées en une seule frontière pour limiter le nombre de calculs de propagation de fronts de vague. Les chemins parcourus par les robots montrent une bonne répartition spatiale des ro-

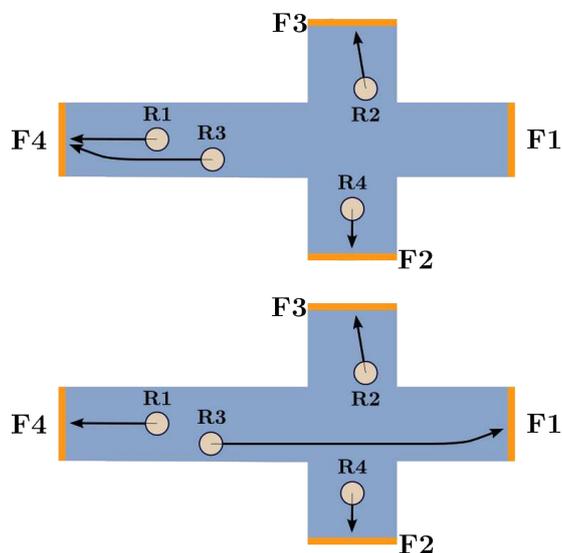


Figure 14. Comparaison des assignations avec les algorithmes *Position Minimum* (en haut) et *Glouton* (en bas)

bots dans l'environnement, c'est-à-dire une absence de redondance de visite des pièces éloignées du hall d'entrée.

Des mesures de temps d'exploration de cette arène ont été menés avec les algorithmes *Position Minimum* et *Frontière la plus proche*. Le graphique donné en figure 16 montre, comme attendu, que *Position Minimum* permet une exploration totale de l'environnement plus rapidement qu'avec *Frontière la plus proche*. L'arène étant petite pour le nombre de robots utilisés (jusqu'à 4), on constate que les écarts se resserrent dès 3 robots. Le système est par ailleurs impacté par les conflits spatiaux engendrés par un nombre croissant de robots. En effet, les robots peuvent être amenés à se gêner aux croisements par exemple. *Position Minimum* crée toutefois moins de situations de gêne. En effet, en se répartissant mieux dans l'environnement, les situations de conflit sont évitées. Ainsi, lorsque nous augmentons le nombre de robots le temps d'exploration reste pratiquement constant tandis qu'avec *Frontière la plus proche* les conflits entre les robots entraînent une augmentation du temps d'exploration. Nous envisageons d'intégrer ces contraintes de conflit de navigation dans l'allocation des frontières pour la poursuite de ce travail. Notons enfin que lors de ces expériences le temps mesuré est celui nécessaire aux robots pour explorer l'environnement mais aussi pour retourner à leur point de départ.



Figure 15. Photo de l'arène d'exploration et carte et trajectoires résultant de l'exploration de l'arène avec 3 robots

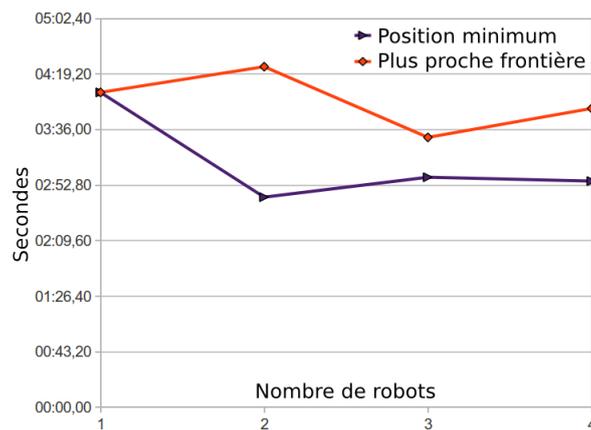


Figure 16. Résultat des expériences d'exploration de l'arène en variant le nombre de robots de 1 à 4 (moyenne sur 4 expériences). L'unité en ordonnée est le temps en minutes et secondes et en abscisse le nombre de robots

6. Conclusion

Nous avons abordé dans ce papier le problème de l'exploration multirobot d'un environnement inconnu. Nous avons proposé un nouvel algorithme pour l'assignation des frontières que découvrent les robots. Celui-ci repose sur la notion de position vers une frontière, déterminée par le nombre de robots plus proches de cette frontière. Les robots sont alors répartis sur les frontières selon leurs meilleures positions, plutôt qu'en fonction des distances aux frontières. L'évaluation des positions est fondée sur la lecture directe du coût en distance dans le champ de potentiels construit depuis chaque frontière. Des mesures en simulation et sur des robots ont permis de montrer que cette approche est plus efficace en temps d'exploration que les algorithmes *Glouton* et *Frontière la plus proche* mais la différence dépend de la topologie de l'environnement. De plus, sa complexité est inférieure aux approches classiques fondées sur l'algorithme *Glouton*. Enfin, la décision est décentralisée, elle ne nécessite pas de communiquer entre robots d'autres informations que les cartes individuelles et les coordonnées des robots. Les perspectives de ce travail concernent le développement d'une carte hybride topologique et métrique qui devrait permettre de limiter l'échange d'information concernant l'environnement et de réduire le coût des calculs de distance, mais aussi les conflits de navigation.

Remerciements

Ce travail a bénéficié du financement de l'ANR CAROTTE, projet Cartomatic.

Bibliographie

- Barraquand J., Langlois B., Latombe J. (1992). Numerical potential field techniques for robot path planning. *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 22, n° 2, p. 224–241.
- Burgard W., Moors M., Stachniss C., Schneider F. (2005, jun.). Coordinated multi-robot exploration. *Robotics, IEEE Transactions on*, vol. 21, n° 3, p. 376–386.
- Duckett T., Nehmzow U. (1997). Experiments in evidence based localisation for a mobile robot. In *Proceedings of the aisb workshop on spatial reasoning in mobile robots and animals, manchester, uk*.
- Elfes A. (1989, Jun). Using occupancy grids for mobile robot perception and navigation. *Computer*, vol. 22, n° 6, p. 46–57.
- Franchi A., Freda L., Oriolo G., Vendittelli M. (2009, 04). The sensor-based random graph method for cooperative robot exploration. *IEEE/ASME Trans. on Mechatronics*, vol. 14, n° 2, p. 163–175. (Winner of the IEEE RAS ICYA Best Paper Award 2010.)
- Howard A. (2006). <http://ijr.sagepub.com/content/25/12/1243.abstract> Multi-robot Simultaneous Localization and Mapping using Particle Filters. *The International Journal of Robotics Research*, vol. 25, n° 12, p. 1243–1256.
- Mantaras R. L. D., Amat J., Esteva F., Lopez M., Sierra C. (1997). Generation of unknown environment maps by cooperative low-cost robots. In *Proceedings of the 1st international conference on autonomous agents*, p. 164–169. ACM Press.

- Simmons R., Apfelbaum D., Burgard W., Fox M., D. an Moors, Thrun S., Younes H. (2000). Coordination for multi-robot exploration and mapping. In *Proceedings of the aaai national conference on artificial intelligence*. Austin, TX, AAAI.
- Thrun S., Liu Y. (2005). Multi-robot slam with sparse extended information filers. In *International journal of robotics research*, p. 254–266. SAGE.
- Yamauchi B. (1997). A frontier-based approach for autonomous exploration. In *Cira '97: Proceedings of the 1997 ieee international symposium on computational intelligence in robotics and automation*, p. 146. Washington, DC, USA, IEEE Computer Society.
- Yamauchi B. (1998). Frontier-based exploration using multiple robots. In *Agents '98: Proceedings of the second international conference on autonomous agents*, p. 47–53. New York, NY, USA, ACM.
- Zlot R. M., Stentz A. T., Dias M. B., Thayer S. (2002, May). Multi-robot exploration controlled by a market economy. In IEEE (Ed.), *IEEE international conference on robotics and automation*, vol. 3, p. 3016–3023.