



HAL
open science

Stratégie d'exploration multi-robot 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 multi-robot fondée sur les champs de potentiels artificiels. Journées Francophones sur les Systèmes Multi-Agents (JFSMA), Oct 2011, Valenciennes, France. inria-00598608

HAL Id: inria-00598608

<https://inria.hal.science/inria-00598608>

Submitted on 25 Oct 2011

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 multi-robot fondée sur les champs de potentiels artificiels

A. Bautin^a O. Simonin^a F. Charpillet^a
antoine.bautin@inria.fr olivier.simonin@loria.fr francois.charpillet@inria.fr

^a Equipe MAIA
INRIA - LORIA - Université Henri Poincaré

Résumé

Le contexte de ce travail est le problème de l'exploration multi-robot 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 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 en simulation sont présentées et montrent la validité et l'efficacité de notre approche. Nous évaluons l'efficacité de notre algorithme en comparant, en simulation, ses performances avec les approches existantes.

Mots-clés : Exploration multi-robot ; Allocation de tâches décentralisée ; Navigation par champs de potentiels artificiels

Abstract

The context of this work is multi-robot 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 minima 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 demonstrated the validity of the frontier allocation approach.

Keywords: Multi-robot exploration ; Decentralized multi-robot task allocation ; Artificial potential field navigation method

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 multi-robot 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 et de communiquer avec les robots. Nous utilisons une représentation discrète de la carte 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 donc 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 intraitable pour des flottilles de taille importante, 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 coût 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 inexplorée
- 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 :

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

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 donc 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 en bas 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 :

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

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 coût, l'assignation de frontière peut être effectuée soit

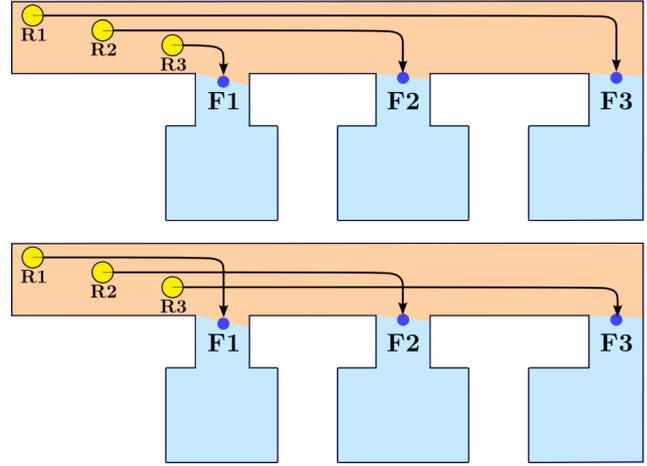


FIGURE 1 – Illustration du critère 3 : le coût total des 2 assignations est identique mais le temps pour réaliser l'exploration complète sera inférieur dans l'assignation du bas.

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 [5] ou le suivi de mur [3]. En mono-robot, comme en multi-robot, une technique admise comme efficace est l'exploration successive des frontières créées entre les zones non-explorées et les zones explorées accessibles [7]. 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 multi-robot d'exploration de frontière [8]. 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 ré-

sultat. 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.

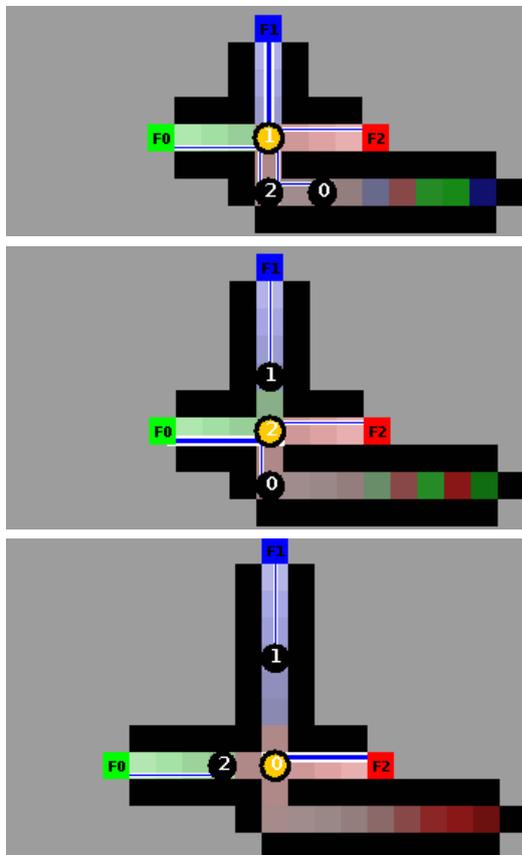


FIGURE 2 – Coordination implicite avec l'algorithme frontière la plus proche. Description du haut vers le bas : quand le premier robot $R1$ atteint l'intersection des couloirs il découvre 3 frontières équidistantes et choisit aléatoirement la frontière $F1$, quand $R2$ atteint l'intersection, il reste 2 frontières plus proches équidistantes et choisit aléatoirement $F0$ entre $F0$ et $F2$, quand le 3^{ème} robot $R0$ arrive sur l'intersection une seule frontière est plus proche de lui et il choisit donc $F2$.

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

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

fin

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

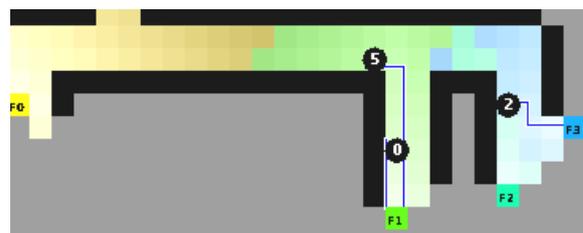


FIGURE 3 – Résultat de l'assignation avec l'algorithme Frontière la plus proche. Les robots ne prennent pas en compte la position des autres robots. Les robots 5 et 0 sont assignés à la même frontière.

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 itère 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 coût, 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 [2] [6] [9].

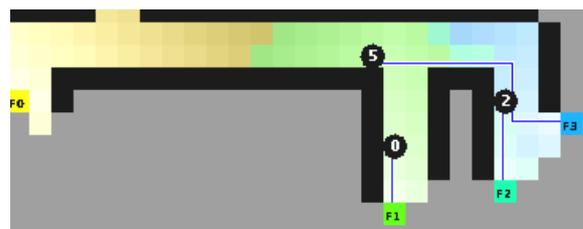


FIGURE 4 – Illustration de l'allocation de frontière avec l'algorithme glouton

Algorithme 2 : Glouton (version décentralisée)

Entrées : \mathcal{C} Matrice de coût**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 $i, j =$
 $\operatorname{argmin} \mathcal{C}_{ij} \quad \forall \mathcal{R}_i \in \mathcal{R}, \forall \mathcal{F}_j \in \mathcal{F}$
 $\alpha_{ij} = 1$
 $\mathcal{R} = \mathcal{R} \setminus \mathcal{R}_i$
 $\mathcal{F} = \mathcal{F} \setminus \mathcal{F}_j$
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. [2] utilisent l'algorithme glouton en intégrant un critère d'importance à chaque frontière. Ils utilisent le concept d'utilité, celui-ci vise à réduire l'importance de l'exploration de frontières proches de celles déjà assignées. En effet, un robot observera le voisinage de la frontière à laquelle il est assigné. Ainsi, l'utilité de l'exploration d'une frontière est inversement proportionnelle à sa distance aux autres frontières dont elle n'est pas séparée par un obstacle. 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 $O(n^2m)$.

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 frontière 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 [6]. 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 [9].

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". 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 F0 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 coût. Pour la calculer, nous construisons un champ de potentiel sans minimum local depuis chaque frontière (algorithme de front de vague [1]). 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 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.

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 vague [4]

- 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 sur un environnement contenant une cellule

1. par occupé nous entendons occupé par un obstacle et non par un robot
2. distance du chemin parcouru depuis la source

frontière. Dans les figures 2, 3, 4 et 6 les cellules sont colorées par le gradient provenant de la frontière la plus proche.

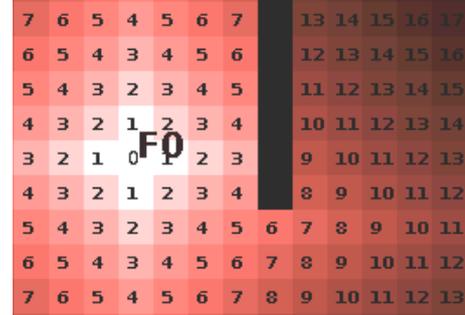


FIGURE 5 – Résultat du champ de potentiels calculé avec l'algorithme 3.

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 (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.

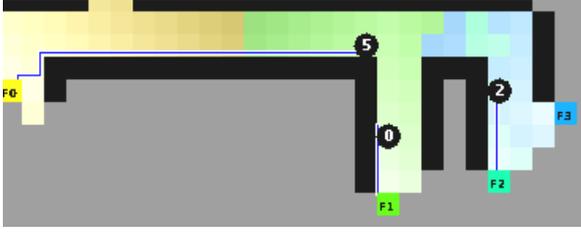


FIGURE 6 – Résultat de l’assignation avec l’algorithme 4 Position Minimum.

Algorithme 4 : Position Minimum

Entrées : \mathcal{C} matrice de coût

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

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

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

fin

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

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

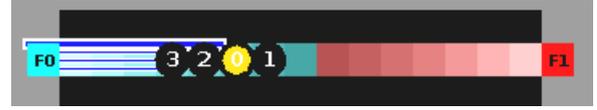
Cette section analyse et compare les performances de l’algorithme proposé.

5.1 Analyse des comportements

En simulant ce modèle, nous avons observé une meilleure répartition des agents qu’avec l’algorithme de frontière la plus proche 1. Plusieurs robots peuvent aller vers la même frontière mais en se répartissant en groupes de taille égale. Cette situation est illustrée par les figures 6 et 7.

La figure 7 illustre un problème typique avec l’allocation par Frontière la plus proche. Un groupe de robots est proche d’une frontière, une autre frontière se trouvant plus loin dans une direction opposée. Tous les robots seront assignés à la même frontière tandis qu’avec l’algorithme Position Minimum le groupe sera séparé en deux groupes de tailles égales.

Par ailleurs, un comportement dynamique pertinent émerge qui tend à séparer les robots groupés. Deux robots qui se suivent pour atteindre une même frontière se séparent dès que le premier s’éloigne d’une frontière orpheline qui



(a) Frontière la plus proche : tous les robots sont assignés à la même frontière



(b) Position Minimum : les robots sont également répartis entre les 2 frontières

FIGURE 7 – Différence entre l’allocation de frontières par l’algorithme Frontière la plus proche et Position Minimum

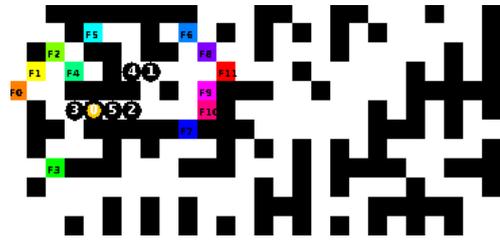


FIGURE 8 – Exemple d’environnement de type labyrinthe

capte ainsi le second robot. En effet il est alors seul en première position.

5.2 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 son voisinage parfaitement. Les expériences ont été réalisées sur des environnements différents comme des labyrinthes générés aléatoirement dont un exemple est illustré par la figure 8 ou de type bâtiment comme la section d’hôpital (figure 9). Les expériences se terminent lorsque les agents ont construit la totalité de la carte. Des vidéos montrant le déroulement de l’exploration sur ces environnements sont disponibles à l’adresse <http://www.loria.fr/~bautin>.

Les figures 10 et 11 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, Glouton et Position Minimum (algorithmes 1, 2 et 4). Sur les 2 environnements, nous observons que les algorithmes Glouton et Position Minimum

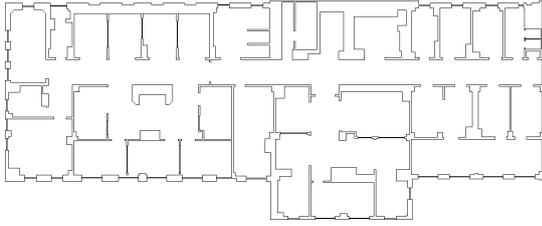


FIGURE 9 – Environnement section d’hôpital de Player/Stage

sont plus performants que l’algorithme Plus proche frontière, écart d’autant plus marqué que le nombre de robots augmente (jusqu’à 20%). L’algorithme proposé, Position Minimum, apparaît légèrement meilleur que le Glouton, ceci est plus significatif sur l’environnement section d’hôpital. De plus, la comparaison des complexités est nettement à l’avantage de Position minimum qui est en $O(nm)$ contre $O(n^2m)$ pour l’algorithme Glouton.

6 Conclusion

Nous avons abordé dans ce papier le problème de l’exploration multi-robot 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 ont permis de montrer que cette approche est plus efficace en temps d’exploration que les algorithmes Glouton et Plus proche frontière. De plus, sa complexité est inférieure aux approches classiques fondées sur l’algorithme Glouton. Enfin, notre approche est décentralisée et nécessite moins de communications que l’algorithme Glouton. Les perspectives de ce travail concernent le déploiement de l’approche avec des robots réels. Nous envisageons également de comparer l’algorithme à des approches exploitant la notion d’utilité, même si celle-ci induit des coûts calculatoire et de communication supplémentaires.

Remerciement

Ce travail a bénéficié du financement de l’ANR Carotte -projet Cartomatic-.

Références

- [1] J. Barraquand, B. Langlois, and J.-C. Latombe. Numerical potential field techniques for robot path planning. In *Advanced Robotics, 1991. 'Robots in Unstructured Environments', 91 ICAR., Fifth International Conference on*, pages 1012 –1017 vol.2, jun. 1991.
- [2] W. Burgard, M. Moors, C. Stachniss, and F.E. Schneider. Coordinated multi-robot exploration. *Robotics, IEEE Transactions on*, 21(3) :376 – 386, jun. 2005.
- [3] Tom Duckett and Ulrich Nehmzow. 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, 1997.*, 1997.
- [4] Steven M LaValle. *Planning Algorithms*. Cambridge University Press, 2004.
- [5] R. Lopez De Mantaras, J. Amat, F. Esteva, M. Lopez, and C. Sierra. Generation of unknown environment maps by cooperative low-cost robots. 1997.
- [6] R. Simmons, D. Apfelbaum, W. Burgard, M. Fox, D. an Moors, S. Thrun, and H. Younes. Coordination for multi-robot exploration and mapping. In *Proceedings of the AAAI National Conference on Artificial Intelligence*, Austin, TX, 2000. AAAI.
- [7] Brian Yamauchi. A frontier-based approach for autonomous exploration. In *CIRA '97 : Proceedings of the 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation*, page 146, Washington, DC, USA, 1997. IEEE Computer Society.
- [8] Brian Yamauchi. Frontier-based exploration using multiple robots. In *AGENTS '98 : Proceedings of the second international conference on Autonomous agents*, pages 47–53, New York, NY, USA, 1998. ACM.
- [9] Robert Michael Zlot, Anthony (Tony) Stentz, M Bernardine Dias, and Scott Thayer. Multi-robot exploration controlled by a market economy. In IEEE, editor, *IEEE International Conference on Robotics and Automation*, volume 3, pages 3016 – 3023, May 2002.

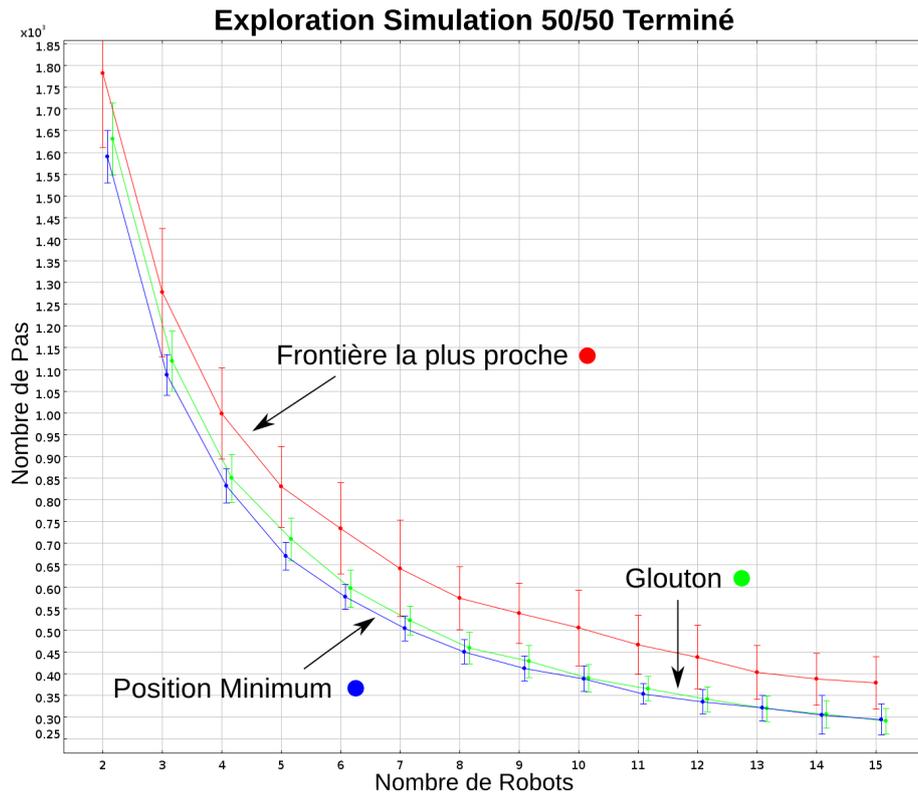


FIGURE 10 – Résultat de la simulation de l’exploration d’un environnement de type labyrinthe

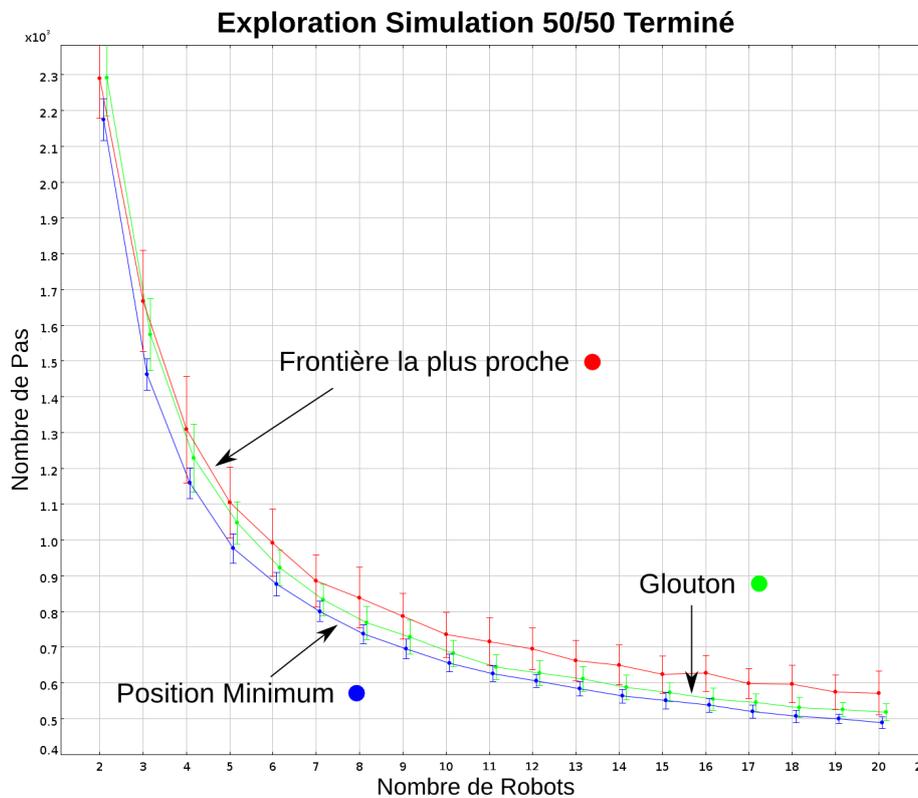


FIGURE 11 – Résultat de la simulation de l’exploration de l’environnement section d’hôpital