

Exploration en 3D par des robots désorientés : tu montes en bas ou tu descends en haut ?

Quentin Bramas¹, Stéphane Devismes², Anaïs Durand³, Pascal Lafourcade³
et Anissa Lamani¹

¹ Université de Strasbourg, CNRS, ICUBE

² Université de Picardie Jules Verne, MIS, UR 4290

³ Université Clermont Auvergne, CNRS UMR 6158, LIMOS

Nous nous intéressons à l'exploration d'un environnement 3D par un petit nombre de robots ayant des capacités très limitées. Ces robots ont une vision à portée bornée et des capacités d'orientation limitées. Précisément, ils doivent explorer perpétuellement une grille 3D bien qu'ils n'aient pas de système de coordonnées commun. Cependant, ils s'accordent sur une chiralité commune. Ils exécutent le même algorithme de manière synchrone et sont équipés de lumières capables de changer de couleur. En outre, chaque robot dispose d'un panel réduit de couleurs et ces couleurs constituent les seules informations pouvant être mémorisées par le robot et perçues par les robots alentour. Dans ce cadre restreint, notre problème consiste à assurer que les robots visitent infiniment souvent et de manière exclusive chaque nœud d'une grille 3D. Notre but est de proposer des solutions mobilisant le moins de ressources possibles. Dans ce contexte, nous montrons qu'avec une visibilité à un saut, trois robots sont nécessaires et suffisants pour résoudre l'exploration perpétuelle exclusive d'une grille 3D. L'algorithme proposé nécessite cependant cinq couleurs. Nous proposons aussi une solution utilisant uniquement une couleur (l'optimal) et cinq robots avec une visibilité à deux sauts.

Mots-clés : robots mobiles, grille 3D finie, algorithme distribué, exploration exclusive perpétuelle.

1 Introduction

Contexte. Cet article traite de l'exploration perpétuelle d'un environnement par un essaim de robots mobiles autonomes équipés de capteurs de visibilité, d'actionneurs de mouvement et de lumières pouvant prendre différentes couleurs. Dans ce problème, l'environnement est un espace divisé en lieux (par exemple, les pièces d'une maison) qui doivent être visités infiniment souvent par au moins un robot.

La littérature sur l'exploration par ce type de robots, appelés robots *lumineux*, s'est jusqu'ici concentrée sur des environnements 2D comme les grilles finies [2] ou infinies [3]. Or, une application naturelle de ce problème est la surveillance d'un bâtiment à l'aide d'une patrouille de robots ou de drones autonomes. Ainsi, nous proposons de nous intéresser à une structure 3D simple : une grille 3D (finie), où les nœuds représentent les lieux à visiter et les arêtes la possibilité d'aller d'un lieu à un autre. Cette grille permet de modéliser, par exemple, un bâtiment en forme de parallélépipède rectangle.

Nos robots lumineux opèrent de manière *synchrone*. Cependant, ils ont des capacités très faibles. Tout d'abord, la portée de leur vision est bornée. Ensuite, seul un nombre constant de couleurs est disponible pour leurs lumières. À part les lumières, les robots n'ont pas de mémoire persistante et aucun moyen de communiquer. Par ailleurs, les couleurs de leurs lumières constituent l'unique moyen de les distinguer. Enfin, ils disposent d'un système d'orientation très faible : les robots n'ont *pas de système de coordonnées commun* (c.-à-d. qu'il n'y a *a priori* pas d'accord sur les orientations Nord-Sud et Est-Ouest), mais ils s'accordent sur une *chiralité commune*. Dans le contexte 3D, la chiralité désigne la capacité d'un robot à distinguer deux côtés d'une symétrie orthogonale par rapport à un plan. La chiralité commune permettra ainsi aux robots de s'accorder sur le côté choisi en présence d'une telle symétrie. Formellement, cela signifie que les robots ont un système de coordonnées arbitraire mais orienté de manière directe.

Contribution. Cet article est un résumé étendu de [1] dont l'objectif était de proposer des algorithmes permettant de résoudre l'*exploration perpétuelle exclusive d'une grille 3D* en minimisant les ressources

utilisées, c.-à-d. en utilisant un nombre minimum de robots lumineux avec un nombre minimum de couleurs différentes ainsi que des capacités très faibles de vision (portée bornée) et d'orientation (ici, uniquement la chiralité). L'exploration perpétuelle d'une grille 3D consiste à assurer que chacun des ces nœuds soit visité infiniment souvent par au moins un robot. L'exclusivité impose de surcroît que l'exploration soit réalisée en n'ayant jamais deux robots occupant le même nœud ni ne traversant la même arête simultanément.

Nous avons tout d'abord étudié le problème sous l'hypothèse d'une vision à un saut (c.-à-d., à distance 1 dans la grille), l'optimal. Sous cette hypothèse, nous avons montré que 3 robots sont nécessaires et suffisants pour résoudre l'exploration perpétuelle exclusive de la grille 3D. Notez que nous avons obtenu la condition suffisante en proposant un algorithme utilisant uniquement 5 couleurs. Nous avons ensuite proposé une solution optimale en nombre de couleurs utilisées, c.-à-d., un algorithme utilisant uniquement une couleur, mais sous l'hypothèse d'une vision à deux sauts ; ce dernier algorithme nécessite seulement 5 robots.

Plan. Le modèle utilisé par nos robots est décrit succinctement dans la section 2. Ensuite, par manque de place, nous résumons les idées principales de notre preuve d'impossibilité (section 3) puis nous décrivons uniquement notre premier algorithme (section 4). Nous concluons dans la section 5 avec quelques perspectives. Les détails omis sont disponibles dans la version longue de cet article [1].

2 Modèle

Nous considérons un essaim de n robots lumineux évoluant dans une grille 3D de taille $x \times y \times z$, c.-à-d., un graphe non-orienté $G = (V, E)$ où $V = \{(i, j, k) : i \in [0..x - 1], j \in [0..y - 1], k \in [0..z - 1]\}$ et $E = \{(i, j, k), (i', j', k') \in V^2 : |i - i'| + |j - j'| + |k - k'| = 1\}$. Ces coordonnées sont utilisées uniquement pour le raisonnement : elles sont inconnues des robots.

Les robots effectuent une infinité de *rondes* durant lesquelles chaque robot exécute un cycle « *Regarder-Calculer-Se Déplacer* » de manière synchrone. À chaque ronde, chaque robot *regarde* autour de lui : il obtient une *vue* du sous-graphe des nœuds situés à distance (de Manhattan) au plus $\Phi \geq 1$ de lui. À partir de cette vue, le robot *calcule* sa nouvelle couleur et sa destination, et *se déplace* vers cette destination en changeant la couleur de sa lumière (*n.b.*, le robot peut choisir de rester immobile et/ou garder la même couleur). Dans une vue, les nœuds sont, le cas échéant, étiquetés avec les couleurs des robots les occupant. Dans les vues et dans les traces d'exécutions, nous matérialisons l'absence de nœuds par une bordure extérieure constituée de nœuds virtuels colorisés en noir. Les bordures extérieures sont appelées *murs*.

En outre, la vue est orientée selon un système de coordonnées ego-centré choisi par un *adversaire* à chaque ronde, ce qui modélise l'absence d'orientation globale. Ainsi, l'orientation d'un robot r peut être différente d'une ronde à l'autre. Précisément, pour assurer une chiralité commune, l'adversaire est restreint dans ses choix : il peut appliquer sur chacun des axes passant par r une rotation prise dans $\{0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}\}$ (la rotation choisie peut-être différente pour chacun des axes).

L'ensemble des couples (*position, couleur*) de tous les robots à la ronde t , définit la *configuration* du système à la ronde t . Un algorithme est décrit par un ensemble de configurations initiales possibles, un ensemble de couleurs \mathbb{C} et un ensemble de règles. L'ensemble des configurations initiales est décrit par un *motif local*, c.-à-d., un placement relatif des robots les uns par rapport aux autres avec leurs couleurs initiales ; ce qui augmente la flexibilité par rapport à une solution avec une seule configuration initiale possible. Une règle associe une vue à une destination, quelle que soit son orientation. Cela signifie qu'un algorithme ne contient au plus qu'une règle pour chaque vue, quelle que soit l'orientation. Un algorithme utilisant n robots doit réaliser l'exploration exclusive perpétuelle dans toutes grilles 3D de taille $x \times y \times z$ avec $x \geq n, y \geq n$ et $z \geq n$ et ce à partir de toute configuration où le motif local initial est placé et orienté de manière quelconque.

3 Résultat d'impossibilité

Théorème 1. *Dans notre modèle, il n'existe pas d'algorithme permettant de réaliser l'exploration perpétuelle dans toutes grilles 3D de taille au moins $n \times n \times n$ avec $n \leq 2$ robots et $\Phi = 1$, quel que soit le nombre constant de couleurs disponibles.*

Pour démontrer ce théorème, nous procédons par contradiction : on suppose qu'un tel algorithme existe. Un robot est dit *isolé* s'il ne voit ni un mur, ni un autre robot.

Le cas $n = 1$ est trivial. En effet, dans une grille 3D d'au moins $4 \times 4 \times 4$, un robot finit nécessairement par atteindre un nœuds u où (1) il est isolé et (2) au moins une des destinations possibles v le garde isolé.

Or, lorsqu'un robot est isolé, il ne peut discriminer les destinations possibles. Donc, s'il décide de bouger, sa destination est choisie par l'adversaire, qui peut ainsi le maintenir isolé en restant pour toujours sur les nœuds u et v , contradiction.

Considérons maintenant le cas $n = 2$. Tout d'abord, comme les robots disposent d'un nombre constant de couleurs $|C|$, il n'y a qu'un nombre borné, B , de règles possibles quelle que soit la grille à explorer. Ainsi, si les deux robots voyagent sans voir un mur durant B rondes, ils vont forcément effectuer une suite de mouvements périodiques. Par ailleurs, lorsqu'ils se déplacent sans voir un mur, ils restent groupés (sinon ils deviennent isolés), leur vue est donc symétrique par rapport à une symétrie de rotation (dont l'axe passe par les deux robots). Les seuls mouvements non ambigus possibles sont donc des mouvements en ligne droite. Les autres mouvements, correspondant aux quatre autres destinations symétriques, sont ambigus dans le sens où l'adversaire peut décider la direction prise par les robots.

L'ensemble de ses propriétés nous a permis de montrer que les robots ne peuvent pas s'éloigner de plus de $4B$ des murs. En effet, toute exécution atteint nécessairement une configuration où les deux robots sont à distance au plus $2B$ de deux murs et à partir de cette configuration, ils peuvent s'éloigner d'un mur pour "longer" l'autre mur, mais il ne peuvent pas s'éloigner de ce dernier après l'avoir longé sur une distance arbitraire : soit ils décident de s'en éloigner avant $2B$ rondes, soit ils continuent de le longer jusqu'à atteindre un autre mur. Dans les deux cas, les déplacements se font en ligne droite et ils restent ainsi à distance au plus $4B$ des murs de la grille.

Ainsi, en considérant une grille suffisamment grande (par exemple $10B \times 10B \times 10B$), il y a des nœuds (par exemple au milieu de cette grille) qui ne sont soit jamais visités soit seulement un nombre fini de fois (au début de l'exécution par exemple), une contradiction qui établit notre théorème.

4 Algorithme

Nous présentons maintenant l'algorithme, appelé Vone_5^3 , qui résout l'exploration pour $n = 3$, $|C| = 5$ et $\phi = 1$. D'après le théorème 1, Vone_5^3 est optimal en nombre de robots. Pour faciliter la compréhension de la description ci-dessous, nous invitons le lecteur à visualiser les animations fournies en ligne [4]. La grille 3D peut être vue comme un bâtiment à plusieurs étages. L'idée générale de l'algorithme proposé est d'explorer la grille 3D étage par étage, comme illustré à la figure 1.

Sur un étage donné, deux robots sont en charge d'explorer l'étage ligne par ligne tandis que le troisième robot est utilisé comme une balise pour garder trace de la direction de l'exploration : il désignera soit la ligne suivante à explorer sur l'étage, soit l'étage suivant à explorer lorsque les robots ont fini d'explorer l'étage courant. Ainsi, l'algorithme définit à l'aide des couleurs trois rôles principaux pour les robots : Leader (la couleur L), Suiveur (la couleur F) et Balise (les autres couleurs). La balise peut prendre trois couleurs différentes A, B et C pour distinguer différentes situations. Notez que les robots peuvent changer de rôle plusieurs fois durant l'exécution de l'algorithme. Par manque de place, nous ne présentons pas les règles de l'algorithme. Cependant, ces dernières sont disponibles dans la version originale de l'article [1] ainsi que dans l'animation en ligne.

Initialement, les robots sont alignés et ont respectivement les couleurs C, C et A. Un des deux robots ayant la couleur C doit être adjacent aux deux autres. Ce motif peut être placé arbitrairement sur la grille 3D.

À partir d'une configuration initiale, les robots se déplacent tout d'abord vers un mur. Une fois le mur atteint, ils se coordonnent entre eux afin d'atteindre un type particulier de configurations, noté C_p , à partir duquel l'exploration effective débute. Autrement dit, le préfixe initial conduisant à une configuration C_p ne se produit qu'une seule fois. Ensuite, le système passe périodiquement par les configurations C_p et tous les nœuds sont visités entre deux occurrences d'une même configuration pendant l'exécution.

Une configuration est de type C_p si les robots sont situés sur deux lignes adjacentes ℓ_i et ℓ_{i+1} de la grille 3D telles que sur ℓ_i il y a deux robots adjacents de couleur F et L respectivement à distance 2 et 3 du même mur et sur ℓ_{i+1} il y a un seul robot de couleur A et adjacent à ce mur. La grille 2D contenant ℓ_i et ℓ_{i+1} est appelé *étage*. La stratégie utilisée pour explorer un étage donné est similaire à celle de [6] : deux robots (le leader et le suiveur) se déplacent et explorent une ligne donnée tandis que le troisième (la balise) reste inactif à côté d'un mur de la grille pour indiquer la prochaine ligne à explorer. Lors de l'exploration d'une ligne

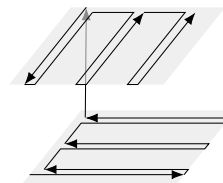


FIGURE 1 : Trajet effectué par les robots pendant l'exploration.

donnée, les deux robots s'éloignent dans un premier temps du mur où se trouve la balise jusqu'à atteindre le mur opposé puis font demi-tour pour revenir sur la même ligne. Ils procèdent ensuite au changement de ligne avec l'aide de la balise. La figure 2 présente les séquences de configuration lors du changement de ligne.

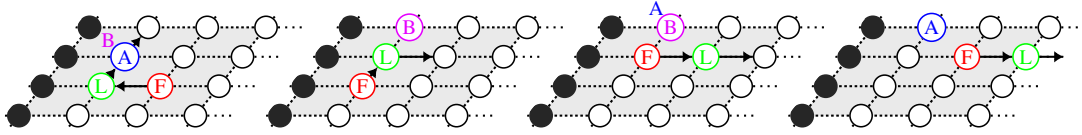


FIGURE 2 : Séquence de configurations lors du changement de ligne sur un étage donné.

Une fois l'étage exploré, les robots passent à l'étage suivant à l'aide de la balise. À noter que le processus utilise la chiralité afin de faire en sorte que les robots explorent les étages dans une direction donnée tant qu'ils n'ont pas atteint le dernier étage. La figure 3, présente la séquence de configuration lors de ce processus. Remarquez que la dernière ligne de l'étage courant est explorée après l'initiation du changement d'étage.

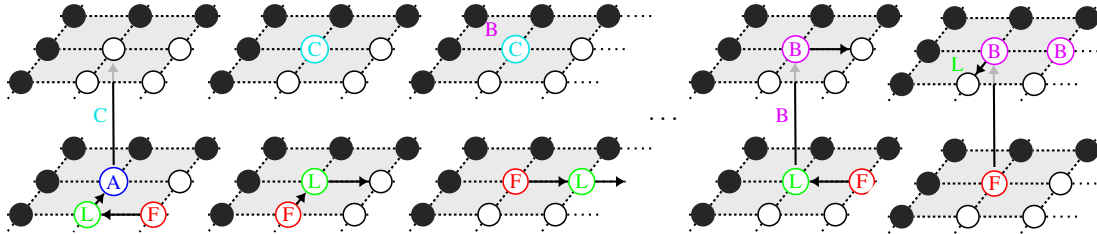


FIGURE 3 : Séquence de configurations lors du changement d'étage.

Lorsque les robots atteignent le dernier étage, appelé toit, ils l'explorent de la même manière à l'exception des couleurs utilisées lors du changement de ligne. Cela est fait pour s'assurer que les robots restent sur le toit sans ajouter de couleurs supplémentaires. Une fois le toit exploré, les robots initient l'exploration de la grille 3D dans le sens inverse. Tous les détails sont disponibles dans [1].

Par manque de place, l'algorithme n'utilisant qu'une couleur, cinq robots, et une vision à deux sauts, n'est pas décrit ici. Mais une animation et les règles de l'algorithme sont disponibles en ligne [5].

5 Conclusion

La perspective immédiate de ce travail est d'étudier l'optimalité en nombre de robots de notre solution (non présentée ici) utilisant cinq robots monochromes avec une vision à deux sauts. Ensuite, nous allons nous intéresser à la recherche d'algorithme pour grille 3D n'utilisant pas la chiralité. Notre expérience des grilles 2D nous suggère que l'optimalité nécessite dans ce cas un robot supplémentaire. Enfin, nous pensons que nos algorithmes peuvent être modifiés automatiquement et avec un surcoût acceptable pour fonctionner sous l'hypothèse asynchrone.

Références

- [1] Quentin Bramas, Stéphane Devismes, Anaïs Durand, Pascal Lafourcade, and Anissa Lamani. Beedroids : How luminous autonomous swarms of uavs can save the world? In *FUN 2022*, pages 7 :1-7 :21, 2022.
- [2] Quentin Bramas, Stéphane Devismes, and Pascal Lafourcade. Optimal Exclusive Perpetual Grid Exploration by Luminous Myopic Opaque Robots with Common Chirality. In *ICDCN 2021*, pages 76-85, 5-8 January 2021.
- [3] Quentin Bramas, Pascal Lafourcade, and Stéphane Devismes. Finding water on poleless using melomaniac myopic chameleon robots. In *FUN 2021*, pages 6 :1-6 :19, 2021.
- [4] Q. Bramas et al. Animation du premier algorithme. <https://bramas.fr/algote12023-3dgrid-animations-1>.
- [5] Q. Bramas et al. Animation du second algorithme. <https://bramas.fr/algote12023-3dgrid-animations-2>.
- [6] Arthur Rauch, Quentin Bramas, Stéphane Devismes, Pascal Lafourcade, and Anissa Lamani. Optimal exclusive perpetual grid exploration by luminous myopic robots without common chirality. In *NETYS 2021*, pages 95-110, 2021.