Un jour sans fin

Stéphane Devismes, Pascal Lafourcade

Un jour sans fin

Stéphane Devismes[1] et Pascal Lafourcade[2]

[1] *Université Grenoble Alpes, CNRS UMR 5104, VERIMAG*
[2] *Université Clermont Auvergne, CNRS UMR 6158, LIMOS*

Cet article est un résumé de travaux [4] portant sur l'exploration exclusive perpétuelle d'une grille finie par un petit nombre de robots aux capacités faibles. Ces robots sont opaques, sans orientation commune et ont une vision à portée limitée. Cependant, ils savent distinguer leur droite de leur gauche, exécutent le même algorithme de manière synchrone et sont équipés de lumières avec un nombre réduit de couleurs. Mise à part cette lumière, les robots n'ont aucune mémoire permanente et aucun moyen de communiquer. Par ailleurs, les couleurs de leurs lumières constituent l'unique moyen de les distinguer. Dans ce cadre, nous étudions l'optimalité en termes de portée, de nombre de robots et de couleurs utilisées. En supposant une portée optimale (1 saut), nous proposons un algorithme optimal en nombre de robots (2) et de couleurs (3). Nous donnons deux autres algorithmes pour une portée à deux sauts : l'un utilise 3 robots et une seule couleur (l'optimal) ; l'autre réalise le meilleur compromis entre le nombre de robots (2) et le nombre de couleurs (2).

Mots-clefs : robots mobiles, grille finie, algorithme distribué, exploration exclusive perpétuelle.

1 Introduction

Nous étudions les essaims de robots autonomes équipés de capteurs de visibilité, d'actionneurs de mouvement et de lumières pouvant prendre différentes couleurs. Nous supposons que ces robots, dits *lumineux*, opèrent de manière *synchrones*. Cependant, ils ont des capacités très faibles. Tout d'abord, leur portée de visibilité est bornée. Ensuite, ils sont *opaques*, c.-à-d., qu'un robot ne peut voir un autre robot si un troisième se situe entre les deux. Le nombre de couleurs disponibles pour les lumières est constant. À part ces 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, 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 ont la *même chiralité* (c.-à-d. une même notion de droite et de gauche). On étudie comment un nombre réduit de ces robots peut réaliser l'*exploration exclusive perpétuelle*. Dans ce problème, l'espace est divisé en lieux qui doivent être visités par les robots. Ainsi, on considère un espace discret fini défini par un graphe (ici une grille), où les nœuds représentent les lieux à visiter et les arêtes la possibilité d'aller d'un lieu à un autre. L'exploration perpétuelle consiste alors à assurer que chaque nœud du graphe soit visité infiniment souvent par au moins un robot. L'exclusivité impose de surcroît que l'exploration soit réalisée sans que jamais deux robots occupent le même nœud ni ne traversent la même arête simultanément.

Contribution. Nous étudions l'optimalité en termes de portée, de nombre de robots et de couleurs utilisées pour résoudre l'exploration exclusive perpétuelle d'une grille finie. Précisément, lorsque la portée est optimale (un saut), nous donnons un algorithme optimal en nombre de robots (2) et de couleurs (3). Avec une portée à deux sauts, nous avons deux algorithmes : l'un utilise trois robots et une seule couleur (l'optimal) ; l'autre réalise le meilleur compromis entre les nombres de robots (2) et de couleurs (2).

État de l'art. L'exploration a été étudiée dans des topologies variées : anneaux [1], arbres [6], grilles finies [5, 2] et infinies [3], ... Les deux solutions pour des grilles finies [5, 2] réalisent l'exploration (avec terminaison [5] ou perpétuelle [2]) sans utiliser de lumière mais avec une visibilité infinie et sans opacité.

Plan. Le modèle utilisé par nos robots est résumé dans la section 2. Nous justifions nos bornes inférieures en section 3. Nous décrivons les algorithmes réalisant ces bornes en section 4. Par manque de place, d'autres résultats de [4] ne sont pas présentés. Ils concernent l'obtention de la terminaison avec quelques couleurs supplémentaires et la version asynchrone du 1er algorithme obtenue en augmentant la portée à deux sauts.

2 Modèle

Une configuration de n robots dans une grille finie est représentée par l'ensemble des couples *(position, couleur)* de chacun 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* une couleur et une 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 (*cf.*, la figure 1). 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 peut être différente d'une ronde à l'autre. Un algorithme est décrit par un ensemble de configurations initiales, un ensemble de couleurs *Cl* 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. Nos algorithmes doivent résoudre l'exploration exclusive perpétuelle à partir de toutes grilles pouvant contenir le motif local initial placé et orienté de manière quelconque. Une règle associe une vue locale à une destination, quelle que soit son orientation. Cela signifie qu'un algorithme ne contient qu'au plus une règle pour chaque vue, quelle que soit l'orientation.

3 Résultats d'impossibilité

Impossibilité pour $n = 1$. Considérons un robot R avec une portée Φ quelconque et un nombre fini de couleurs disponibles. Dans une grille de taille au moins $(2\Phi+3) \times (2\Phi+3)$, R ne peut plus distinguer les destinations possibles lorsqu'il atteint le milieu de la grille et qu'il ne voit plus les bords. Dans ce cas, soit il reste immobile, soit il se déplace. Dans le second cas, à la ronde suivante, il se retrouve dans une situation comparable (ils ne voient toujours pas les bord) et puisque l'orientation est choisie par l'adversaire, ce dernier peut forcer le robot à revenir à sa position initiale. Ainsi, l'adversaire peut faire échouer l'exploration.

Impossibilité pour $n = 2$ et $|Cl| = 1$. On considère encore une grande grille. Par contradiction, l'un des deux robots finit par atteindre le milieu (tous les nœuds devant être visités) et dans ce cas soit il reste immobile, soit l'adversaire peut le contraindre à alterner entre deux positions proche du milieu jusqu'à ce que le deuxième robot soit à distance au plus Φ. Avec une grille suffisamment grande, lorsque cela arrive, aucun des deux robots ne voit les bords de la grille. Les deux robots sont donc dans une situation symétrique et comme dans le cas précédent, soit les deux robots restent immobiles, soit l'adversaire peut choisir leur orientation à chaque ronde pour les faire alterner entre deux positions indéfiniment.

Impossibilité pour $n = 2$ et $|Cl| = 2$ et $\Phi = 1$. Comme précédemment, on considère une grille très grande. Pour pouvoir se déplacer, les robots doivent rester groupés à cause de la portée à un saut. Ensuite, depuis le centre de la grille, les deux robots doivent nécessairement se déplacer en ligne droite en ayant deux couleurs différentes et fixes. Lorsqu'ils atteignent un bord de la grille, les deux cas suivants sont possibles.

Après un nombre constant de mouvements, ils repartent dans l'autre sens, mais dans ce cas leurs couleurs et positions relatives correspondent à une rotation de 180 degrés par rapport à la traversée précédente. Ainsi, lorsqu'ils atteignent le bord opposé, ils se comportent de la même manière. En conséquence, ils se déplacent pour toujours dans une sous-grille de largeur bornée et ainsi l'exploration échoue.

Dans le deuxième cas, les robots se déplacent en restant à distance bornée du bord jusqu'à atteindre un coin de la grille. Deux sous-cas sont alors possibles. Soit les robots repartent dans le sens inverse toujours à distance bornée du bord et ainsi voyagent indéfiniment entre deux bords. Soit ils quittent le coin par le bord opposé et dans ce cas, ils font indéfiniment le tour de la grille en restant à distance bornée du bord, abandonnant ainsi le milieu.

Impossibilité pour $n = 3$ et $|Cl| = 1$ et $\Phi = 1$. On montre que dans une grille très grande, les trois robots peuvent se retrouver loin des bords. À partir de là, l'adversaire peut choisir l'orientation des robots de telle manière qu'au moins un robot soit hors de portée des deux autres après au plus deux rondes, tout en assurant qu'aucun robot ne voit les bords de la grille pendant ce temps. Depuis cette dernière situation, l'adversaire peut contraindre les trois robots à s'isoler, toujours loin des bords, après quoi il peut les maintenir bloqués pour toujours dans un petit périmètre sans qu'ils voient ni les autres, ni les bords.

4 Algorithmes

Notations. Dans les règles et exemples présentés ci-dessous, la limite extérieure de la grille est symbolisée par un *mur* représenté par des nœuds noirs. Nous adoptons en plus une écriture compacte pour les règles et

les exemples afin de regrouper plusieurs situations semblables en une seule : le mouvement d'un robot dans une situation donnée peut être indépendant du fait qu'il soit voisin d'un nœud vide ou d'un mur ; dans ce cas, nous regroupons les deux cas en un seul à l'aide un nœud hachuré.

Principe. Nos algorithmes suivent, à peu près, le même principe. Les robots se déplacent toujours ensemble en serpentin suivant un chemin hamiltonien. Par serpentin, nous entendons que les robots se déplacent en ligne droite d'un bord à l'autre de la grille, puis changent de ligne lorsqu'ils atteignent le mur. Les changements de lignes suivent la même direction jusqu'à ce que les robots atteignent un coin de la grille après avoir longé le bord. Dans ce cas, ils changent de phase lors du virage, puis reprennent l'exploration, toujours en serpentin, mais dans la direction opposée. Ainsi, ils alternent à l'infini ces deux différentes phases et ainsi visitent tous les nœuds infiniment souvent.

Ci-dessous, nous présentons rapidement les spécificités de nos trois algorithmes. Pour plus de détails, nous invitons le lecteur à visualiser les animations fournies en ligne [†].

Algorithme pour $n = 2$, $|Cl| = 3$ et $\Phi = 1$. Dans cet algorithme, deux robots, appelés *leader* et *suiveur*, sont initialement groupés et le restent tout au long de leurs déplacements. La couleur du suiveur, S, est fixe. La lumière du leader alterne entre deux couleurs G et D (pour Gauche et Droite). La couleur initiale du leader, G ou D, n'a pas d'importance. Le fait que les lumières du leader et du suiveur diffèrent permet de se déplacer en ligne droite entre les murs en utilisant les règles de la figure 1 : l'algorithme du suiveur consiste uniquement à suivre le leader ; et entre deux murs, le leader se déplace dans la direction opposée au suiveur. Pour se déplacer en serpentin, le groupe doit alternativement tourner à gauche et à droite à chaque fois qu'il atteint un mur. La couleur du leader désigne alors la direction à prendre par le groupe. Après le virage, le leader change sa couleur (précisément lorsqu'il quitte le mur, *cf.* la figure 2) afin de se rappeler de tourner dans l'autre sens au prochain mur. La figure 3 explicite les mouvements des robots lorsque le leader atteint un mur avec la couleur D. Le groupe a un comportement analogue lorsque le leader a la couleur G. Lorsque le groupe atteint la dernère ligne de la grille, il effectue une dernière traversée le long du mur (en suivant les mêmes règles, c.-à-d. celles données en figure 1) puis atteint un coin où il lui est impossible de tourner dans le sens indiqué par la couleur du leader, c'est à ce moment précis que s'effectue le changement de phase avec les règles de la figure 4. La couleur du leader va changer deux fois lors du virage ce qui permettra de repartir en serpentin dans la direction opposée à la phase précédente : lorsque le leader atteint un mur et que la destination désignée par sa couleur est un mur, il choisit la direction opposée et change de couleur, il reprend ensuite l'algorihme précédent, en particulier en changeant à nouveau de couleur en quittant le mur. Les figures 5 et 6 montrent les deux changements de phases possibles en fonction de la couleur du leader.

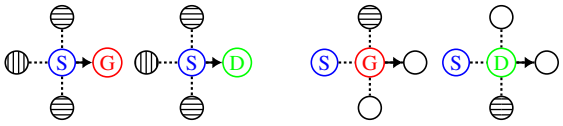

FIGURE 1: Déplacement en ligne droite.

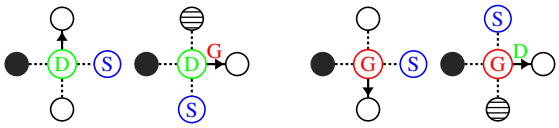

FIGURE 2: Virage devant un mur du leader D et G.

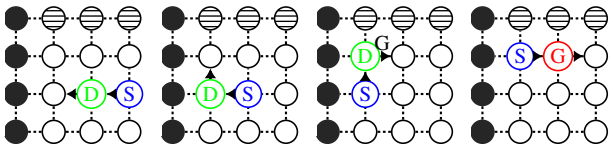

FIGURE 3: Virage lorsque le leader a la couleur D.

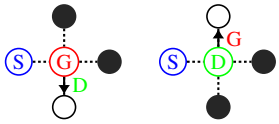

FIGURE 4: Changement de phase au coin.

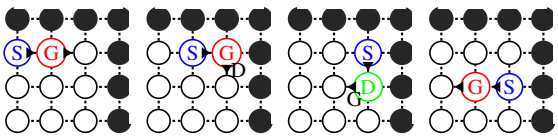

FIGURE 5: Changement de phase.

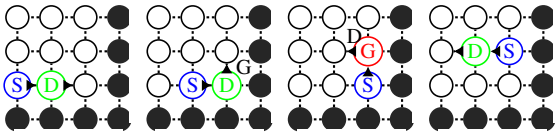

FIGURE 6: Changement de phase avec l'autre couleur.

†. https://doi.org/10.5281/zenodo.3947756

Algorithme pour $n = 2$, $|Cl| = 2$ et $\Phi = 2$. Cet algorithme se sert de la distance à deux sauts pour permettre aux deux robots de ne pas rester « coller » en permanence. Ainsi, la distance relative entre les deux robots est utilisées pour coder le sens du virage lorsque le groupe atteint un mur et on économise ainsi une couleur. Les deux robots ont donc deux couleurs constantes différentes, *S* et *L*, leur permettant de distinguer le rôle respectif, leader ou suiveur. Ils partent initialement groupés. Pour se déplacer en ligne droite entre les murs, ils avancent comme précédemment, en maintenant leur distance relative qui alterne entre un et deux après chaque virage. Lorsque le groupe atteint un mur séparé, il tourne à droite comme illustré en figure 7. Notez que les deux robots quittent le mur collés, ce qui leurs indiquera de tourner à gauche face au mur de l'autre côté. Une fois de l'autre côté, ils suivront la même idée que pour l'algorithme précédent sauf qu'ils devront se séparer à la sortie du virage.

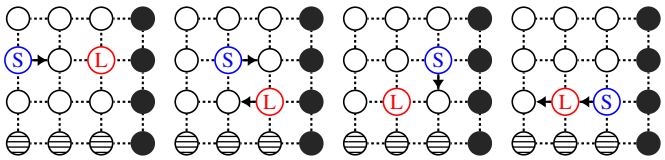

FIGURE 7: Virage à droite.

Le reste de l'algorithme concerne le changement de phase qui est lui aussi obtenu en adaptant les règles de l'algorithme précédent. Ces règles sont présentées dans l'article [4] et visibles avec nos animations en ligne [†].

Algorithme pour $n = 3$, $|Cl| = 1$ et $\Phi = 2$. Pour obtenir le nombre de couleurs optimales, il nous faut un robot de plus afin d'utiliser la forme du groupe pour coder la droite et la gauche. Nous utilisons deux formes particulières : le « L » et le « < » (« < » est le motif initial). Ces deux formes sont maintenues durant la traversée d'un bord à l'autre et sont ensuite inversées lors des virages. Les figures 8 et 9 illustrent les deux virages possibles lorsque le groupe arrive face à un mur. Elles montrent en particulier comment le groupe repart sur une ligne adjacente dans la formation autre que celle qu'il avait en arrivant. Par exemple, dans la figure 8, les robots arrivent en formation « L » tournent à gauche et repartent en formation « < » en ayant exploré tous les positions dans le virage. Dans la seconde séquence, les robots arrivent en « < » et repartent en « L » sur une autre ligne. Le reste de l'algorithme, notamment la gestion le changement de phase à la fin du serpentin, est disponible dans l'article [4] et visible avec nos animations en ligne [†].

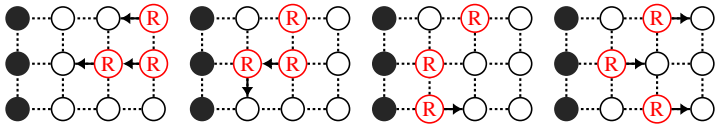

FIGURE 8: Le groupe en formation « L » atteint un mur.

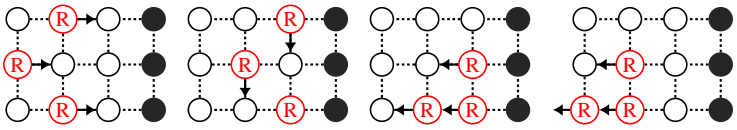

FIGURE 9: Le groupe en formation « < » atteint un mur.

Références

[1] L. Blin, A. Milani, M. Potop-Butucaru, and S. Tixeuil. Exclusive perpetual ring exploration without chirality. In N. A. Lynch and A. A. Shvartsman, editors, *DISC 2010*, pages 312–327, 2010.

[2] F. Bonnet, A. Milani, M. Potop-Butucaru, and S. Tixeuil. Asynchronous exclusive perpetual grid exploration without sense of direction. In *OPODIS 2011*, pages 251–265, 2011.

[3] Q. Bramas, S. Devismes, and P. Lafourcade. Finding Water on Poleless using Melomaniac Myopic Chameleon Robots. In *FUN 2020*, pages 6 :1–6 :19, 2021.

[4] Q. Bramas, P. Lafourcade, and S. Devismes. Optimal exclusive perpetual grid exploration by luminous myopic opaque robots with common chirality. In *ICDCN '21*, pages 76–85, 2021.

[5] S. Devismes, A. Lamani, F. Petit, P. Raymond, and S. Tixeuil. Terminating exploration of a grid by an optimal number of asynchronous oblivious robots. *The Computer Journal*, pages 132–154, 2021.

[6] P. Flocchini, D. Ilcinkas, A. Pelc, and N. Santoro. Remembering without memory : Tree exploration by asynchronous oblivious robots. *Theor. Comput. Sci.*, pages 1583–1598, 2010.