Qui ne se ressemble pas s'assemble

Quentin Bramas, Anissa Lamani, Sébastien Tixeuil

Qui ne se ressemble pas s'assemble[†]

Quentin Bramas[1] et Anissa Lamani[1] et Sébastien Tixeuil[2]

[1] *Université de Strasbourg, ICUBE, CNRS, France*
[2] *Sorbonne Université, CNRS, LIP6, France*

Nous nous intéressons au problème du rendez-vous de deux robots autonomes à capacités réduites. Il a été démontré dans la littérature que le problème n'admet pas de solution déterministe dans le modèle semi-synchrone si les robots sont amnésiques et que leur système de coordonnées est symétrique.

Dans cet article, nous montrons qu'il suffit que les robots admettent des unités de distance différentes pour que le problème devienne soluble.

Mots-clefs : Réseaux de robots, rassemblement, désaccord

1 Introduction

Nous nous intéressons aux cohortes de robots déployées dans un univers continu [2]. Les robots sont dotés de capteurs visuels leur permettant de voir leur environnement ainsi que d'actionneurs de mouvement leur permettant de se déplacer. Les robots sont *anonymes* (ils ne peuvent pas être distingués), *désorientés* (chacun possède son propre système de coordonnées) et *amnésiques* (ils ne peuvent pas se rappeler des actions entreprises précédemment). Ils opèrent en exécutant des cycles comprenant des phases de *vision*, *calcul* et *déplacement*. Plus précisément, dans un cycle, un robot commence par observer son environnement et voit ainsi les positions des autres robots (phase de vision) dans son système de coordonnées. En se basant sur cette observation, le robot calcule une destination (phase de calcul) et enfin se déplace vers la destination calculée (phase de déplacement). La grande majorité des recherches effectuées dans ce modèle se concentre sur la caractérisation des hypothèses minimales pour résoudre des tâches distribuées. Dans la plupart des cas, ces hypothèses sont étroitement liées au degré de synchronisation entre les robots. Trois principaux modèles de synchronisation ont été considérés : le modèle entièrement synchrone (FSYNC) qui oblige *tous* les robots à exécuter leurs cycles simultanément, le modèle semi-synchrone (SSYNC) qui permet à un sous-ensemble non vide de robots d'exécuter leur cycle simultanément, et enfin le modèle asynchrone (ASYNC) qui ne fait aucune hypothèse sur la vitesse relative de chaque robot ou de chaque phase.

Parmi les problèmes de base considérés, nous retrouvons le problème du *rendez-vous*, où deux robots doivent se retrouver en un temps fini, dans un même endroit, qui n'est pas connu à l'avance. Il a été démontré que dans le cas FSYNC, le problème était solvable [2], tandis que dans le cas SSYNC, le problème n'admettait pas de solutions déterministes sans hypothèses supplémentaires.

L'une des principales raisons de ce résultat d'impossibilité est que les deux robots peuvent avoir des vues symétriques initialement. Si les deux robots sont activés d'une manière synchrone, des mouvements symétriques sont alors exécutés leur permettant ainsi de rejoindre un point de rendez-vous en même temps. Cependant, lorsqu'un seul robot est activé (comme c'est possible en SSYNC), un des deux mouvements symétriques est seulement exécuté, empêchant les robots de se rencontrer dans un endroit précis (ils ne font que converger l'un vers l'autre).

Dans ce papier, nous étudions la possibilité de considérer l'influence des unités de distance des systèmes de coordonnées des robots pour résoudre le problème du rendez-vous dans le cas SSYNC. Plus précisément, étant donnés deux robots désorientés ayant des unités de distance différentes, le rendez-vous devient possible sans hypothèses supplémentaires (robots désorientés, amnésiques et déterministes). Notons que le cas où les robots ont les mêmes unités de distance est connu pour ne pas admettre de solution déterministe.

[†] Ce travail a été financé en partie par le projet ANR project SAPPORO, ref. 2019-CE25-0005-1 et par le projet ANR ESTATE, ref. ANR-16-CE25-0009-03.

2 Modèle

Nous considérons deux robots, évoluant dans un espace euclidien à deux dimensions. Les robots sont modélisés sous forme de points et sont supposés être anonymes (ils ne peuvent pas être distingués), uniformes (ils exécutent le même algorithme) et amnésiques (ils ne peuvent pas se souvenir des actions passées).

Soit Z un repère global. Une *configuration* au temps t, notée C_t, est un ensemble $\{r_1, r_2\}$ contenant les positions des deux robots dans Z au temps t. A noter que r_i, $i = 1, 2$, désigne à la fois un robot et sa position en $\mathbb{R}^2$ dans le repère Z. Les robots ne connaissent pas Z, cependant, chaque robot r_i a son propre système de coordonnées Z_{r_i} centré sur la position actuelle de r_i. Nous supposons des robots *désorientés* (ils ne s'accordent sur aucun axe) qui ont des unités de distance différentes. Soit ρ le rapport entre la plus grande et la plus petite unité de distance des robots *i.e.*, $unit_2 = \rho \cdot unit_1$, avec $unit_2 > unit_1$. Pour un robot r, d_r désigne la distance entre les deux robots dans son propre système de coordonnées. Ainsi, pour r et r', les deux robots de notre système, $d_r = \rho d_{r'}$ ou $d_r = \frac{d_{r'}}{\rho}$. Pour des raisons de simplicité, dans la suite de l'article, r_1 désignera le robot avec la plus grande unité de distance et r_2 désignera l'autre robot *i.e.*, $d_1 < d_2$ (où on écrit abusivement d_i au lieu de d_{r_i}, pour $i = 1, 2$). Bien sûr, un robot *ne sait pas* si il a la plus grande unité de distance.

Les robots opèrent en cycles qui comprennent trois phases : *vision*, *calcul* et *déplacement*. Plus précisément, à chaque instant, un robot activé observe son environnement et détermine la position de l'autre robot dans son système de coordonnées ego-centré. Sur la base de cette observation, le robot calcule une destination ou décide de rester inactif. Enfin, le robot se déplace vers la destination calculée (le cas échéant) en suivant une trajectoire rectiligne. Nous supposons que les mouvements sont *non rigides i.e.* un robot peut être arrêté n'importe où le long de sa trajectoire vers sa destination calculée. Cependant cela est possible après avoir parcouru au moins une distance strictement positive, arbitrairement petite mais fixe, égale à δ. La valeur de δ est commune aux deux robots mais elle n'est pas connue des robots.

Dans la configuration C, la *vue locale* d'un robot r_i, notée $\mathcal{V}_{r_i}$ est le résultat de la phase de vision. Plus précisément, lorsqu'un robot r_i observe son environnement, il observe la position de l'autre robot dans son propre système de coordonnées Z_{r_i} (translaté par $-r_i$ pour que r_i soit toujours au centre). Un algorithme A est une fonction qui associe des vues locales à des destinations. Lorsque r_i est activé au temps t, l'algorithme A détermine la destination de r_i dans son propre système de coordonnées Z_{r_i} en se basant sur $\mathcal{V}_{r_i}$.

Nous considérons le modèle SSYNC où à chaque instant t, un sous-ensemble non vide de robots est activé par une entité externe appelée *scheduler* ou *adversaire* . Les robots activés exécutent alors leur cycle vision-calcul-déplacement d'une manière synchrone. Nous supposons que le scheduler est équitable *i.e.*, chaque robot est activé infiniment souvent. Une exécution $\mathcal{E} = (C_0, C_1, \dots)$ d'un algorithme A est une séquence de configurations où C_0 est une configuration initiale, et chaque configuration C_{t+1} est obtenue à partir de C_t après l'exécution de A par les robots activés par l'adversaire.

3 Solution pour $\rho = 2$

Pour commencer, considérons le cas simple où $\rho = 2$. Dans ce cas, si les robots ne sont pas déjà regroupés, on appelle le *niveau* $l_i \in \mathbb{Z}$ d'un robot r_i, l'unique entier tel que $d_i \in [2^{-l_i}, 2^{-l_i+1})$. Comme par hypothèse $d_2 = 2d_1$, nous savons que $l_2 = l_1 - 1$. Nous présentons l'algorithme 1 qui résout le problème du rendez-vous quand $\rho = 2$.

Algorithm 1: Algorithme pour le cas $\rho = 2$, exécuté par le robot r.

Si $l_r \equiv 0 \mod 2$ **alors** rester immobile **sinon** aller sur l'autre robot.

Visuellement :

cas où $l_r \equiv 0 \mod 2$

cas où $l_r \equiv 1 \mod 2$

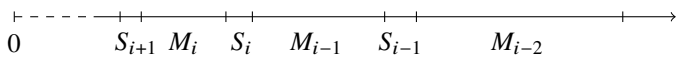


FIGURE 1: Partition de $\mathbb{R}_+^*$ en niveaux

Étant donné que le scheduler est équitable, le robot autorisé à se déplacer finira par exécuter son mouvement. Soit d_t la distance entre les deux robots au temps t. Si $d_t > \delta$ alors, le robot peut ne pas atteindre sa destination. Cependant, la distance entre les deux robots décroît. Ainsi, si le rendez-vous n'est pas effectué, il existe un temps t tel que pour tout $t' > t$, $d_{t'} \leq \delta$. A partir de t, une fois que le scheduler active le robot pouvant se déplacer, le rendez-vous est réalisé. Nous pouvons donc déduire le théorème suivant :

Théorème 1 *L'algorithme 1 résout le problème du rendez-vous en SSYNC quand* $\rho = 2$.

4 Solution pour $\rho \in [\rho_{\min}, \rho_{\max}]$

Nous nous intéressons maintenant au cas général. Nous supposerons toute fois que les robots ont connaissance d'une borne inférieure et d'une borne supérieure sur ρ *i.e.*, $\rho \in [\rho_{\min}, \rho_{\max}]$. La définition des niveaux dans ce cas est un peu plus complexe. Nous définissons deux séquences infinies d'intervalles :

$$\forall i \in \mathbb{Z} \qquad S_i = [\rho_{\min}^{-i}\rho_{\max}^{-i}, \rho_{\min}^{-(i-1)}\rho_{\max}^{-i}) \qquad M_i = [\rho_{\min}^{-i}\rho_{\max}^{-(i+1)}, \rho_{\min}^{-i}\rho_{\max}^{-i})$$

Les ensembles S_i et M_i sont appelés *niveaux*. Notons d'abord que $\bigcup_{i\in\mathbb{Z}} S_i \cup M_i = \mathbb{R}_+^*$ et que les intervalles sont deux à deux disjoints. Ainsi, les séquences forment une partition de $\mathbb{R}_+^*$. La figure 1 présente cette partition. Par convention, on considère que les niveaux sont ordonnés dans l'ordre inverse de leur position sur la droite $\mathbb{R}_+^*$. Ainsi, pour tout $i \in \mathbb{Z}$, M_i (respectivement S_i) est plus grand que $M_{i'}$ (respectivement $S_{i'}$) si $i > i'$. De plus, M_i est plus grand que S_i.

Nous pouvons observer que lorsque la distance d_r observée par un robot r décroît, le niveau de ce dernier augmente. Par abus de langage, nous dirons que r est dans l'ensemble X (ou r a le niveau X) si la distance d_r observée par r est dans X. Les niveaux sont regroupés en quatre ensembles définis comme suit :

$$\mathcal{M}_0 = \bigcup_{i\equiv 0 \mod 2} M_i, \qquad \mathcal{M}_1 = \bigcup_{i\equiv 1 \mod 2} M_i, \qquad \mathcal{S}_0 = \bigcup_{i\equiv 0 \mod 2} S_i, \qquad \mathcal{S}_1 = \bigcup_{i\equiv 1 \mod 2} S_i$$

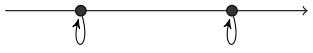
Par convention, les indices sont confondus modulo 2 (par exemple, $\mathcal{M}_{-1} = \mathcal{M}_3 = \mathcal{M}_1$). Soit $\mathfrak{s}(d)$ la plus petite valeur définie comme suit :

$$d\frac{1-\rho_{\min}^{-1}}{2\rho_{\min}^2\rho_{\max}^2} \leq \mathfrak{s}(d) \leq d\frac{1-\rho_{\min}^{-1}}{2} \text{ tel que } \mathfrak{s}(d) \in \mathcal{S}_0 \tag{1}$$

On dit que robot r exécute `Move`($\mathfrak{s}$) lorsque ce dernier se déplace d'une distance $\mathfrak{s}(d_r)$ en direction de l'autre robot. La seconde partie de la définition de $\mathfrak{s}(d)$ garantit que si un robot r exécute `Move`($\mathfrak{s}$) et si l'autre robot a pour destination r (et que chacun atteint sa destination), alors r se retrouve dans $\mathcal{S}_0$. Les inégalités garantissent qu'un tel nombre existe et que si les deux robots exécutent `Move`($\mathfrak{s}$), alors leur niveau augmente d'au plus 1 (ils ne changent pas de niveau ou passent de $\mathcal{M}_i$ à S_{i+1}, mais ne peuvent pas passer de $\mathcal{M}_i$ à $\mathcal{M}_{i+1}$ directement).

Algorithm 2: Le déplacement de r dépend de la distance d entre les deux robots (observée par r) et le côté de la ligne sur lequel r se positionne (à droite ou à gauche) dans son propre système de coordonnées.

$d \in \mathcal{S}_0 \cup \mathcal{S}_1$



$d \in \mathcal{M}_0$

Le robot à droite se déplace à distance $\mathfrak{s}(d)$

$d \in \mathcal{M}_1$

Le robot à gauche se déplace à distance $\mathfrak{s}(d)$

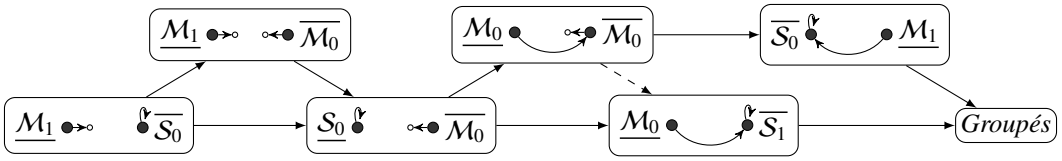


FIGURE 2: Une partie du graphe des configurations et leurs transitions. Le niveau avec une barre au-dessus (resp. au-dessous), est le niveau du robot r_1 (resp. r_2), *e.g.* $\underline{\mathcal{M}_1}$ $\overline{\mathcal{S}_0}$ indique que r_1 se voit à droite avec le niveau $\mathcal{S}_0$ et r_2 à gauche avec le niveau $\mathcal{M}_1$. Les flèches entre les niveaux indiquent le choix de l'algorithme dans ce cas.

Les séquences d'intervalle ont été choisies de telle manière à ce que les deux robots ne peuvent pas être en même temps dans le niveau $\mathcal{S}$ et leurs niveaux ne peuvent pas être très éloignés. En effet, on peut prouver que si $r_1 \in S_i$ alors $r_2 \in M_{i-1}$ et si $r_2 \in S_i$ alors $r_1 \in M_i$ (par hypothèse $d_1 < d_2$).

Ainsi, l'algorithme 2 est défini de telle manière que les deux robots ne soient jamais tous les deux, en même temps, dans un état stationnaire. En exécutant l'algorithme 2, la distance entre les deux robots décroît et il existe donc un temps t à partir duquel les mouvements des robots deviennent rigides. Il suffit par la suite de montrer que quels que soient les niveaux possibles des robots, l'orientation de leur système de coordonnées, et les robots activés (un des deux, ou les deux en même temps), il existe un temps t' tel que le rendez-vous est effectué. Nous avons le résultat suivant :

Théorème 2 *Soit $\rho \in [\rho_{\min}, \rho_{\max}]$, l'algorithme 2 résout le problème du rendez-vous en SSYNC.*

Pour prouver ce résultat, on remarque qu'une configuration est caractérisée par le niveau des robots et par la manière dont ils se perçoivent sur la ligne où ils se situent. Soit ils se perçoivent de façon cohérente (l'un se voit à gauche et l'autre se voit à droite) soit de façon incohérente (les deux se voient à gauche ou les deux se voient à droite).

Dans chaque cas, on connaît le mouvement et donc la prochaine configuration. On peut ainsi calculer le graphe des configurations où un arc représente la possibilité pour les robots d'atteindre une configuration à partir d'une autre. La Figure 2 montre ce graphe pour un sous-ensemble des configurations, dans le cas où les robots perçoivent leur position sur la ligne de façon cohérente. Un arc pointillé signifie que la configuration est atteinte uniquement si un seul robot est exécuté. Il suffit alors de vérifier qu'à partir d'une configuration quelconque, la configuration *Groupés* est forcément atteinte en un temps fini.

Les preuves détaillées sont disponibles dans la version complète du papier [1].

5 Conclusion et perspectives

Nous avons introduit la possibilité d'utiliser les unités de distance pour casser les symétries initiales d'une manière déterministe dans les systèmes de robots amnésiques qui opèrent dans le modèle vision-calcul-déplacement.

La question ouverte qui découle naturellement de ce travail est d'étudier la possibilité de résoudre le problème du rendez-vous dans le modèle ASYNC sans ajouter d'hypothèses ou de contraintes supplémentaires (probabilités, mémoire) autre le fait que les robots ont des unités de distance différentes. Le problème semble cependant difficile même pour le cas $\rho = 2$. En effet, l'algorithme 1 ne résout pas le problème dans le modèle ASYNC étant donné que nous pouvons construire une exécution infinie dans laquelle les robots s'observent alternativement pendant qu'ils se déplacent, et ne peuvent ainsi jamais atteindre l'autre robot.

Références

[1] Quentin Bramas, Anissa Lamani, and Sébastien Tixeuil. The agreement power of disagreement. In Colette Johnen, Elad Michael Schiller, and Stefan Schmid, editors, *Stabilization, Safety, and Security of Distributed Systems - 23rd International Symposium, SSS 2021*.

[2] Ichiro Suzuki and Masafumi Yamashita. Distributed anonymous mobile robots : Formation of geometric patterns. *SIAM Journal on Computing*, 28(4) :1347–1363, 1999.