



HAL
open science

Réveil de robots distribué optimal

Cyril Gavaille, Nicolas Hanusse, Gabriel Le Boudier, Taïssir Marcé

► **To cite this version:**

Cyril Gavaille, Nicolas Hanusse, Gabriel Le Boudier, Taïssir Marcé. Réveil de robots distribué optimal. ALGOTEL 2025 – 27èmes Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications, Jun 2025, Saint Valéry-sur-Somme, France. hal-05029176

HAL Id: hal-05029176

<https://hal.science/hal-05029176v1>

Submitted on 10 Apr 2025

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.



Distributed under a Creative Commons Attribution 4.0 International License

Réveil de robots distribué optimal[†]

Cyril Gavoille¹, Nicolas Hanusse¹, Gabriel Le Bouder¹ et Taïssir Marcé¹

¹LaBRI, Université de Bordeaux, CNRS

Le problème du réveil de robots (*Freeze-Tag Problem*) introduit par Arkin *et al.* (SODA'02) consiste à réveiller n robots placés dans le plan, partant d'un unique robot éveillé. Une fois réveillé, un robot peut à son tour participer au réveil. Ce problème a initialement été formulé comme un problème centralisé, et plusieurs travaux ont permis successivement d'améliorer le temps de réveil dans le pire des cas. Récemment, Bonichon *et al.* [BGHO24] ont proposé un algorithme dont le temps de calcul est linéaire en n et dont le temps de réveil est au plus 4.63ρ , ρ étant la distance maximale entre le robot initialement éveillé et un robot endormi. Dans cet article, nous étudions la version distribuée de ce problème, dans laquelle les robots ont une visibilité limitée à leur disque unité. Nous proposons un algorithme distribué qui permet de réveiller n robots dans le plan sous l'hypothèse que les robots connaissent un majorant ℓ du seuil de connectivité du graphe des positions initiales. Nous montrons que notre algorithme garantit un temps de réveil en $O(\rho + \ell^2 \log(\rho/\ell))$, et que ce temps est asymptotiquement optimal.

Mots-clés : Agents Mobiles, Algorithmique Distribuée, Exploration Collaborative

1 Introduction

Le problème du réveil des robots, en anglais *Freeze-Tag Problem* (FTP), consiste à réveiller le plus rapidement possible un essaim de robots inactifs ou endormis en supposant qu'un robot est initialement actif ou éveillé. Pour réveiller un robot, un robot éveillé doit se déplacer sur la position du robot endormi. Une fois qu'un robot devient actif, il peut aider à réveiller d'autres robots. FTP a été introduit dans un cadre centralisé où les positions des robots endormis sont connues du robot initial.

Dans cet article, nous proposons une version distribuée du FTP : (1) l'emplacement des robots endormis est inconnu ; (2) les robots actifs n'ont de visibilité qu'à distance au plus 1 ; et (3) les robots doivent être co-localisés pour communiquer. Étant donné que le disque de visibilité ne pourrait contenir aucun robot endormi, il peut être nécessaire d'explorer davantage pour localiser d'autres robots endormis.

1.1 État de l'art et contributions

Une solution du FTP est encodée par un arbre de réveil binaire dont les nœuds sont les robots et le parent d'un robot représente le robot qui l'a réveillé. Même pour des cas simples, une solution optimale du FTP (c'est-à-dire un arbre dont le temps de réveil est minimum) ne peut pas raisonnablement être calculée en temps polynomial. Arkin *et al.* [ABF⁺06] ont montré que, même pour les métriques d'étoiles, FTP est NP-complet, et qu'obtenir une $(5/3 - \varepsilon)$ -approximation est NP-difficile pour les métriques générales de graphes. On peut cependant obtenir une approximation constante sur des graphes non pondérés en supposant exactement un robot endormi par nœud [ABG⁺03].

Dans cet article, nous nous concentrons sur la version *géométrique*. Il n'y a pas de restriction sur les déplacements et le graphe sous-jacent est le graphe complet du plan euclidien. Le problème reste cependant NP-difficile [AAJ17]. Lorsque les robots sont localisés dans le disque unité ($\rho = 1$), Najafi Yazdi *et al.* [YBMK15] ont proposé un algorithme linéaire avec un temps de complétion 10.07. Ce résultat a été

[†]Financé par les projets ANR-22-CE48-0001 (TEMPOGRAL) et ANR-24-CE48-7768-01 (ENEDISC).

amélioré par Bonichon *et al.* [BCGH24] pour des normes arbitraires, soit 7.07 dans le cas euclidien, et plus récemment avec 4.63 [BGHO24] également en temps linéaire, la valeur optimale étant conjecturée à $1 + \sqrt{2} \approx 3.83$ [BCGH24].

Une seule variante correspond à la situation où tous les robots ne sont pas connus d'avance [HNP06, BW20]. Dans cette version, chaque robot apparaît à un temps inconnu mais devient instantanément visible de tous à son apparition. Dans [BW20], les auteurs proposent une solution avec un ratio de compétitivité de $1 + \sqrt{2}$ par rapport à l'arbre de réveil optimal où positions et temps d'apparition seraient révélés à l'avance. Dans leur solution, les arêtes de l'arbre de réveil final ne correspondent plus nécessairement à des plus courts chemins mais à des chemins temporels autorisant les robots à revenir plusieurs fois à une même position.

Dans notre cadre, nous ne supposons pas que l'apparition de robots est "gratuite" car les robots ont un rayon de visibilité unitaire. Pour découvrir de nouveaux robots, ils doivent explorer leur environnement. Ainsi, découvrir un robot à distance R peut prendre un temps $\Omega(R^2)$.

1.2 Modèles et hypothèses

Capacité des robots. Le robot *source* est noté s et sa position initiale est $p_s = (0, 0)$. L'ensemble des robots est noté $\mathcal{R} = \{s, r_1, \dots, r_n\}$ où chaque robot r_i est initialement endormi et positionné en p_i et on note \mathcal{P} l'ensemble des positions initiales. Chaque robot possède une lumière indiquant son état : endormi ou réveillé. Les robots réveillés se déplacent à vitesse 1. Des robots co-localisés peuvent se communiquer leurs variables. Chaque robot stocke notamment la position initiale et l'état des robots dont il a connaissance. Ils partagent un même référentiel spatio-temporel issu du robot source.

On suppose que tous les calculs et communications locales prennent un temps négligeable. Pour se synchroniser, les robots peuvent attendre à un point de rendez-vous un certain temps, ou jusqu'à une date fixée. Enfin, chaque robot réveillé exécute son algorithme en parallèle des autres robots.

Dispersion de \mathcal{P} . Nos résultats sont reliés à la distribution de \mathcal{P} . Par exemple, si la distance entre chaque paire de robots est plus grande que 1, chaque robot pourrait avoir l'impression qu'il est seul.

Pour $\ell \geq 0$, le *graphe ℓ -disque* de \mathcal{P} est le graphe géométrique dont les sommets sont \mathcal{P} , et chaque paire $u, v \in \mathcal{P}$ est une arête si la distance $|uv| \leq \ell$. On appelle *seuil de connectivité* de \mathcal{P} , noté $\ell(\mathcal{P})$, la plus petite valeur de ℓ telle que le graphe ℓ -disque de \mathcal{P} est connexe. Enfin, le *rayon* de \mathcal{P} centré en s , noté $\rho(\mathcal{P}, s)$, est la plus grande distance de s aux points de \mathcal{P} .

Contributions. Nous proposons un algorithme distribué qui, étant donné un majorant ℓ sur le seuil de connectivité de \mathcal{P} connus de la source, garantit un temps de complétion $O(\rho + \ell^2 \log(\rho/\ell))$ où ρ est le rayon de \mathcal{P} . D'autre part, nous montrons que ce temps est asymptotiquement optimale car, pour tout algorithme distribué, il est possible de construire une distribution de points nécessitant un temps $\Omega(\rho + \ell^2 \log(\rho/\ell))$. La section suivante donne les éléments essentiels pour comprendre l'algorithme et notre borne inférieure. Une description plus détaillée et complète est disponible dans [GHLBM25].

2 Un algorithme distribué

Notre algorithme $\mathcal{A}_{\text{separator}}$ est basé sur plusieurs opérations distribuées : (1) l'exploration de carrés de largeur R en temps $O(R^2/k + R)$ par une équipe de k robots ; (2) le réveil distribué et la propagation d'arbres de réveil qui se fait en temps $O(R)$; (3) l'exploration de séparateurs géométriques de largeur ℓ ; et (4) le calcul d'échantillonnage de robots dans un carré.

Pour explorer avec k robots, il suffit de partitionner le carré en k rectangles et chaque robot découvre les robots endormis de son rectangle. Les robots se retrouvent ensuite sur un coin ou au centre du carré pour partager leur connaissance. Un des robots réveille les robots endormis du carré en temps $O(R)$.

Le ℓ -séparateur d'un carré S , noté $\text{sep}(S)$, est le bord intérieur de S et de largeur ℓ . Le bord de S formant un séparateur (courbe de Jordan), et par connexité du graphe ℓ -disque, il est clair que s'il existe un robot r

à l'intérieur de S et un robot r' à l'extérieur de S alors il existe un chemin du graphe ℓ -disque entre r et r' passant un robot de $\text{sep}(S)$. L'ensemble des robots de $\text{sep}(S)$ est noté \mathcal{X} .

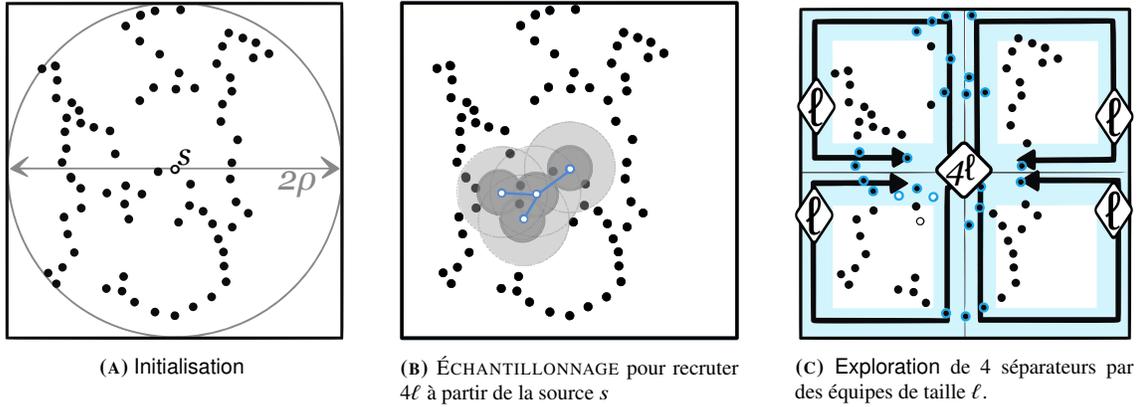


FIGURE 1 : Premières phases de $\mathcal{A}_{\text{Separator}}$; Les positions initiales sont marquées \bullet pour les robots endormis, \circ robots réveillés; Les positions initiales situées dans un séparateur sont cerclées de bleu : \bullet endormi; \circ réveillé

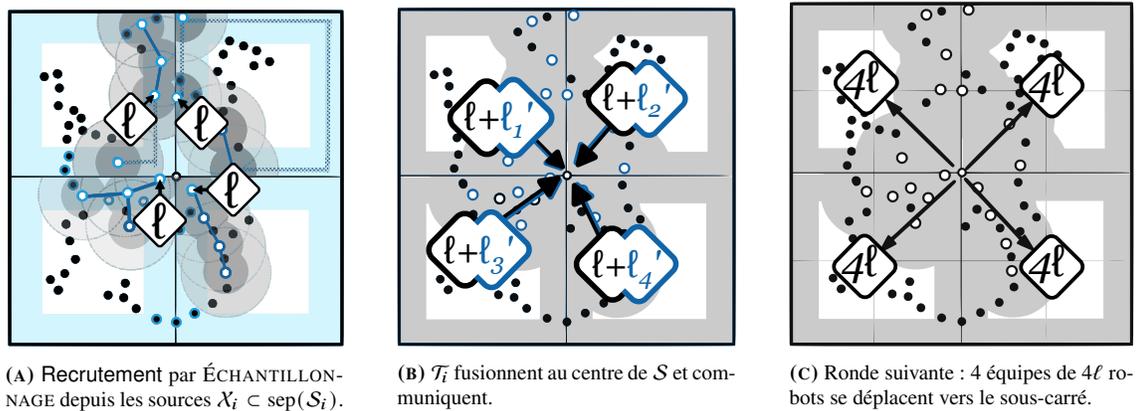


FIGURE 2 : Recrutement et Reorganisation. \circ robots recrutés; zone explorée en gris.

Enfin, un ℓ -échantillonnage de $\mathcal{P} \cap S$ est un ensemble de robots à distance au moins ℓ les uns des autres. Nous avons conçu un algorithme distribué, nommé ÉCHANTILLONNAGE, permettant de construire un ℓ -échantillonnage de taille au plus 4ℓ dans S avec équipe de robots \mathcal{T} à partir des positions de \mathcal{X} . On note A l'ensemble des positions initiales de robots réveillés et connus par une équipe \mathcal{T} . Pour des raisons techniques, la connaissance de A est nécessaire (non détaillé ici).

Dans la Figure 1b, pour $\ell = 1$, le robot source constitue une équipe de 4 robots. Les zones en gris clair sont les zones explorées à rayon 2ℓ et les zones en gris foncé sont de rayon ℓ et ne contiennent qu'au plus un robot échantillonné, par définition. Il se peut que la procédure s'arrête avec moins de points mais cela signifie que tous les robots de S ont été découverts et qu'on peut alors les réveiller directement.

Par soucis de simplification, on suppose dans un premier temps que la source connaît ρ . Comme on le verra plus tard cette hypothèse peut-être supprimée. Le principe de $\mathcal{A}_{\text{Separator}}$ est le suivant : nous procédons en un découpage récursif du premier carré de largeur 2ρ centré en s , en recrutant une équipe de taille 4ℓ (Fig. 1a et 1b). Les robots recrutés explorent les 4 séparateurs correspondant aux régions à distance au plus ℓ des bords de chaque sous-carré (Fig. 1c, zones en bleu clair). Dans chaque sous-carré, nous recrutons une nouvelle équipe de taille $\ell' \leq 4\ell$ avec la procédure ÉCHANTILLONNAGE, constituée de (nouveaux

ou anciens) robots réveillés dans les sous-carrés correspondants (Fig. 2a). Enfin, chaque équipe centralise sa connaissance au centre du carré initial, puis les nouvelles équipes de taille 4ℓ vont au centre de chaque sous-carré (Fig. 2b et 2c). Si l'opération de recrutement a trouvé $\ell' < 4\ell$ robots, alors nous pouvons montrer qu'il n'y a plus besoin de continuer à subdiviser le carré, il suffit de réveiller les robots endormis que l'on vient de découvrir.

Informellement, le temps de complétion se décompose en $O(\log(1 + \rho/\ell))$ rondes, la k -ième ronde prenant un temps $O(\ell^2 + \rho/2^k)$ pour $k > 0$. Voir [GHLBM25] pour l'analyse détaillée.

$\mathcal{A}_{\text{separator}}$

1. Ronde 0 : Initialisation et premier recrutement
 - $\mathcal{S} \leftarrow$ carré de largeur $R = 2\rho$ et centré au robot source s
 - Recruter jusqu'à $4\ell - 1$ robots dans \mathcal{T} en utilisant ÉCHANTILLONNAGE(\mathcal{S}, \mathcal{T})
 - Déplacer \mathcal{T} au centre de \mathcal{S}
2. Ronde $k \geq 1$ pour une équipe \mathcal{T} dans \mathcal{S} :
 - (a) Terminaison
 - Si $|\mathcal{T}| < 4\ell$: Calculer et exécuter un arbre de réveil centralisé dans \mathcal{S} , puis terminer.
 - (b) Partition
 - Partitionner \mathcal{S} (resp. \mathcal{T}) en 4 sous-carrés $\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3, \mathcal{S}_4$ (resp. 4 équipes $\mathcal{T}_1, \mathcal{T}_2, \mathcal{T}_3, \mathcal{T}_4$ de ℓ robots).
 - Pour chaque \mathcal{T}_i , faire en parallèle :
 - (iii) Exploration
 - Explorer collectivement le séparateur $\text{sep}(\mathcal{S}_i)$.
 - Stocker dans X_i les positions initiales des robots de $\text{sep}(\mathcal{S}_i)$, aussi bien ceux encore endormis que ceux précédemment réveillés.
 - (iv) Recrutement
 - Stocker dans A_i les positions initiales des robots réveillés qui sont connus dans \mathcal{S}_i
 - Recruter ℓ'_i robots avec \mathcal{T}_i en utilisant ÉCHANTILLONNAGE($\mathcal{S}_i, \mathcal{T}_i$)
 - Déplacer \mathcal{T}_i au centre de \mathcal{S} .
 - (e) Réorganisation
 - Attendre le retour au centre des 4 équipes \mathcal{T}_i , puis fusionner et partager la connaissance.
 - Réorganiser les robots en équipe \mathcal{T}'_i de taille ℓ'_i en fonction des carrés d'origine \mathcal{S}_i .
 - Pour chaque équipe \mathcal{T}'_i , faire en parallèle :
 - Déplacer \mathcal{T}'_i au centre de \mathcal{S}_i
 - Exécuter Ronde $k + 1$ avec l'équipe \mathcal{T}'_i dans le carré \mathcal{S}_i .

Connaissance de ρ . En l'absence de la connaissance de ρ , il est possible de découvrir une 2-approximation de sa valeur à la fin de la Ronde 0, en parcourant successivement les séparateurs de carrés dont on fait doubler la taille jusqu'à ce que le séparateur d'un des carrés soit vide.

Borne inférieure. Pour démontrer la borne inférieure, nous considérons un algorithme et des valeurs de n, ℓ, ρ fixés. L'idée est de placer des robots endormis dans des régions, disjointes et de rayon $\ell/4$. La localisation exacte du robot dans une région est définie comme le dernier point de la région à être exploré par les robots précédemment réveillés dans l'algorithme. Il y a $\Omega(\rho^2/\ell^2)$ telles régions, et on peut montrer que l'exploration d'une région prend un temps $\Omega(\ell^2/k)$ si k robots sont précédemment réveillés. Pour réveiller tous les robots, il faut explorer toutes les régions, k évolue donc de 1 à $\Omega(\rho^2/\ell^2)$, ce qui conduit à un temps total de $\sum_k \ell^2/k = \Omega(\ell^2 \log(\rho/\ell))$. Voir [GHLBM25] pour plus de détails.

3 Perspectives

Nous avons présenté un algorithme optimal pour une variante distribuée du problème de réveil des robots. Nous pourrions être intéressés par l'ajout d'une contrainte telle que la limitation de la consommation énergétique des robots, en bornant par un seuil B la quantité de déplacement que peut faire chaque robot.

Références

- [AAJ17] Zachary Abel, Hugo A. Akitaya, and Yu Jingjin. Freeze tag awakening in 2D is NP-hard. In *27th Annual Fall Workshop on Computational Geometry (FWCG)*, November 2017.
- [ABF⁺06] Esther M. Arkin, Michael A. Bender, Sándor P. Fekete, Joseph S.B. Mitchell, and Martin Skutella. The freeze-tag problem : How to wake up a swarm of robots. *Algorithmica*, 46 :193–221, 2006.
- [ABG⁺03] Esther M. Arkin, Michael A. Bender, Dongdong Ge, Simai He, and Joseph S.B. Mitchell. Improved approximation algorithms for the freeze-tag problem. In *15th Annual ACM Symposium on Parallel Algorithms and Architectures (SPAA)*, pages 295–303. ACM Press, June 2003.
- [BCGH24] Nicolas Bonichon, Arnaud Casteigts, Cyril Gavoille, and Nicolas Hanusse. Freeze-tag in L_1 has wake-up time five with linear complexity. In Dan Alistarh, editor, *38th International Symposium on Distributed Computing (DISC)*, volume 319 of *LIPICs*, pages 9 :1–9 :16, October 2024.
- [BGHO24] Nicolas Bonichon, Cyril Gavoille, Nicolas Hanusse, and Saeed Odak. Euclidean freeze-tag problem on plane. In *36th Canadian Conference on Computational Geometry (CCCG)*, pages 199–205, July 2024.
- [BW20] Josh Brunner and Julian Wellman. An optimal algorithm for online freeze-tag. In *10th International Conference Fun with Algorithms (FUN)*, volume 157 of *LIPICs*, pages 8 :1–11, September 2020.
- [GHLBM25] Cyril Gavoille, Nicolas Hanusse, Gabriel Le Boudier, and Taïssir Marcé. Distributed freeze tag : a sustainable solution to discover and wake-up a robot swarm. Technical Report 2503.22521v1 [cs.DS], arXiv, March 2025.
- [HNP06] Mikael Hammar, Bengt J. Nilsson, and Mia Persson. The online freeze-tag problem. In *7th Latin American Symposium on Theoretical Informatics (LATIN)*, volume 3887 of *Lecture Notes in Computer Science*, pages 569–579. Springer, March 2006.
- [YBMK15] Ehsan Najafi Yazdia, Alireza Bagheri, Zahra Moezkarimia, and Hamidreza Keshavarz. An $O(1)$ -approximation algorithm for the 2-dimensional geometric freeze-tag problem. *Information Processing Letters*, 115(6-8) :618–622, June 2015.