



FÉLIX QUINTON, CHRISTOPHE GRAND, CHARLES LESIRE

Enchères pour le Maintien des Communications lors de l'Allocation de Tâches pour des Missions Multi-robots

Volume 4, n° 2 (2023), p. 97-122.

DOI not yet assigned

© Les auteurs, 2023.



Cet article est diffusé sous la licence  
CREATIVE COMMONS ATTRIBUTION 4.0 INTERNATIONAL LICENSE.  
<http://creativecommons.org/licenses/by/4.0/>



*La Revue Ouverte d'Intelligence Artificielle est membre du  
Centre Mersenne pour l'édition scientifique ouverte*  
[www.centre-mersenne.org](http://www.centre-mersenne.org)  
e-ISSN : 2967-9672

# Enchères pour le Maintien des Communications lors de l'Allocation de Tâches pour des Missions Multi-robots

Félix Quinton<sup>a</sup>, Christophe Grand<sup>b</sup>, Charles Lesire<sup>b</sup>

<sup>a</sup> AirFrance Group, ITDV OD, CDGRPO Bat 6042 Siège E (3Nord), 45 Rue de Paris, 95747 Roissy Charles de Gaulle (France)

*E-mail* : fequinton@airfrance.fr

<sup>b</sup> ONERA, the French Aerospace Lab, Département Traitement de l'Information et Systèmes, 2 Avenue Edouard Belin, 31000 Toulouse (France)

*E-mail* : christophe.grand@onera.fr, charles.lesire@onera.fr.

---

**RÉSUMÉ.** — Au cours de la dernière décennie, des applications industrielles des systèmes multi-robots ont vu le jour. Parmi ces applications, on trouve des systèmes chargés de missions de pick and delivery, tels que le fameux terminal portuaire autonome de Rotterdam, mais aussi de nombreuses propositions de systèmes chargés de tâches de surveillance. Or, une étape clef de l'exécution de toute mission multi-robots est la résolution du problème d'allocation de tâches au système multi-robots. Ce problème consiste à assigner les tâches de la mission aux robots, et devient un problème d'optimisation combinatoire lorsque l'on associe à la simple allocation des tâches, un objectif d'optimisation représentant les performances du système. Pour résoudre le problème d'allocation de tâches au système multi-robots, les chercheurs ont proposé plusieurs approches. Parmi ces propositions, une classe de méthodes approchées, basées sur les mécanismes d'enchères, a retenu l'attention des chercheurs pour leur capacité à rapidement réallouer les tâches si cela permet d'améliorer l'exécution de la mission. Dans ce papier, nous introduisons un nouveau terme dans l'évaluation des mises dans le cadre d'un protocole d'allocation de tâches par enchères. Ce nouveau terme permet de prendre en compte la connectivité du réseau de communication, qui représente les liaisons de communication entre les membres du système multi-robots. Cet élément est important pour garantir l'efficacité des méthodes basées enchères, puisqu'elles se basent sur le partage d'information entre les membres du système multi-robots pour produire des allocations performantes. Après avoir décrit notre protocole d'enchères, nous avons analysé chacune de ses étapes pour établir des bornes sur sa complexité en temps de calcul et sur la taille des messages échangés au cours de la mission. Nous avons évalué notre méthode dans un scénario de patrouille, et avons montré grâce à des simulations numériques que la préservation des communications améliore la robustesse du système multi-robots face à des événements dynamiques, survenant au cours de la mission de façon imprévisible, et altérant ses conditions d'exécution.

**MOTS-CLÉS.** — Systèmes multi-robots, allocation de tâches, approche basées enchères, mission de surveillance.

---

## 1. INTRODUCTION

Les systèmes multi-robots (SMR) font l'objet de nombreux efforts de recherche depuis une trentaine d'année, et continuent de susciter l'intérêt des chercheurs de nos jours [2, 4, 18, 45]. La pérennité de cet intérêt peut s'expliquer par la diversité des problèmes posés par le déploiement des SMR, et par leur complexité. L'un de ces problèmes est le problème d'Allocation de Tâches au système Multi-Robots (ATMR), qui consiste à attribuer un ensemble de tâches aux robots composant un SMR [15], en optimisant un objectif. Cette modélisation doit être adaptée aux caractéristiques particulières de la mission multi-robots que le SMR exécute. En particulier, l'application des SMR pour des missions de patrouille a été largement étudié [31, 34]. Pour assurer que la mission est exécutée de façon efficace et fiable par le SMR, il faut élaborer une stratégie de coordination, et la résolution de l'ATMR est une étape incontournable de ce processus. L'ATMR étant un problème NP-hard [11], la plupart des recherches visant à le résoudre dans des scénarios en temps réel se concentrent sur des méthodes approchées [27].

En particulier, les approches basées marché ont fait l'objet de nombreux travaux de recherche [15, 27]. Dans ces méthodes, les robots qui composent le SMR sont en concurrence pour obtenir les tâches, qui sont attribuées par le biais d'un mécanisme de marché imitant les transactions économiques du monde réel. Une sous-catégorie d'approches basées marché ayant fait l'objet d'une attention particulière de la part des chercheurs est celle des approches basées enchères (ABE), dans lesquelles les tâches sont vendues grâce à un système d'enchères, dont les commissaires priseurs sont les robots [13, 30, 42, 44]. Les ABE excellent dans la répartition efficace et robuste des tâches simples [7]. Toutefois, les ABE nécessitent un Réseau de Communication (RC) fiable pour que le système de marché fonctionne correctement [29]. Par exemple, si un robot est déconnecté du RC, il peut être incapable de communiquer ses messages contenant ses offres pour les tâches de la mission. Or, un robot n'émettant pas d'offre en réponse à l'annonce d'une enchère n'est pas pris en compte dans la résolution de l'ATMR par une ABE. Par conséquent, une ABE pourrait ne pas attribuer de tâche à ce robot. Cela est d'autant plus vrai lorsque l'on prend en compte le caractère dynamique inhérent à l'exécution de missions de patrouille par un SMR. En effet, le SMR peut rencontrer de nombreux événements dynamiques pendant la mission, entraînant des modifications des paramètres de la mission, et nécessitant que l'ATMR soit résolu en temps réel, c'est à dire, de procéder à des réallocations au cours de la mission. Pour que ces réallocations produisent des solutions efficaces, chaque robot doit disposer localement de données pertinentes et actualisées sur les paramètres de la mission et le comportement de ses coéquipiers, ce qui nécessite que les robots soient capables d'échanger des informations. Une connectivité dégradée du RC entraîne donc une baisse d'efficacité et des vulnérabilités. Pourtant, peu de travaux sont consacrés à l'amélioration du RC dans le SMR.

Dans cet article, nous présentons une méthode d'évaluation des enchères, dont l'objectif est de maintenir la connectivité du RC, permettant aux robots de partager les messages relatifs à l'ABE et aux données partagées aussi largement que possible

au sein du SMR. Les missions de patrouille impliquent généralement des modifications dynamiques du RC, pouvant être causées, par exemple, par l'augmentation de la distance entre les robots. En anticipant ces évolutions, notre méthode consiste à pénaliser les attributions qui divisent le RC en sous-RC réunissant moins de robots, et formant ainsi des sous-équipes ne communiquant pas. D'autre part, elle récompense les attributions qui fusionnent des sous-RC en RC réunissant plus de robots, qui seront alors capables de partager leurs données plus largement grâce à des communications multi-hop. Nous appliquons cette méthode d'évaluation des enchères à une mission de patrouille dans laquelle des événements dynamiques nécessitent des réallocations de tâches. Ces événements représentent des modifications imprévisibles des conditions d'observation au niveau de certaines positions à surveiller pour les besoins de la mission de patrouille multi-robots. Certains des robots composant le SMR ne sont pas capables de répondre à ces événements, et il est donc nécessaire de procéder à des réallocations dynamiques au cours de l'exécution de la mission. Nos résultats montrent que la prise en compte de la connectivité du RC dans le processus d'allocation des tâches améliore la robustesse du SMR face à ces événements dynamiques.

Après une revue de la littérature pertinente, nous exposons la mission étudiée et décrivons le schéma ABE que nous avons utilisé. Ensuite, nous présentons notre méthode d'évaluation des enchères, qui inclut notre terme de maintien de la connectivité du RC. Puis, nous détaillons nos expériences numériques et en discutons les résultats. Cette article se conclut sur un résumé de nos principaux résultats et des propositions de pistes pour les recherches futures.

## **2. REVUE DE L'ÉTAT DE L'ART**

De nombreuses approches ont été proposées, au cours des années, pour résoudre le problème d'allocation de tâches aux systèmes multi-robots. De nombreux auteurs ont appliqué à ce problème des méthodes d'optimisation complète, permettant théoriquement d'atteindre des solutions optimales. Ainsi, l'algorithme Branch-and-bound, représentant les variables de décisions par un arbre, a été utilisé pour l'allocation de tâches [19, 24]. Cet algorithme est également mis à contribution, en plus d'autres optimisations, lorsque le problème d'allocation des tâches est représentée sous la forme d'un problème linéaire en nombre entier [17]. Dans ce cas, les chercheurs utilisent souvent des solvers commerciaux, tels que CPLEX ou Gurobi.

Bien que les approches de résolution complète garantissent l'optimalité des solutions à la convergence de l'algorithme de résolution, elles sont peu adaptée à la plupart des applications concrètes des systèmes multi-robots. En effet, ces derniers évoluent le plus souvent dans des environnements fortement dynamiques et imprévisibles, incompatibles avec les temps de résolution très longs des approches complètes. C'est pourquoi de nombreuses études proposent des algorithmes de résolution approchés pour l'allocation de tâches aux systèmes multi-robots. Parmi ces algorithmes, on trouve notamment des approches méta-heuristiques, telles que le recuit simulé [26, 43], la recherche Tabu [48], ainsi que des algorithmes évolutionnistes,

comme des essais de particules [1] ou des algorithmes de colonies [20]. D'autres algorithmes consistent à répartir les tâches de la mission en paquets [36], et les principales difficultés de ces approches sont de construire ces paquets de façon pertinente [28]. Une autre classe d'algorithmes approchés sont les approches comportementales, qui permettent aux robots de se répartir les tâches sans se concerter, grâce à un ensemble de règles prédéfinies [3]. Ces approches sont donc particulièrement bien adaptées aux missions au cours desquelles les robots sont le plus souvent en autonomie. L'ensemble des comportements des robots peuvent être gérés par des règles comportementales, allant des comportements les plus bas niveaux, tels que les déplacements, à l'allocation des tâches [35]. Par exemple, une règle déclenchant la réallocation des tâches d'un robot après qu'il ait subi une panne permet d'assurer la robustesse du système multi-robots face à ce type d'évènements dynamiques [47].

Enfin, des stratégies de résolution du problème d'allocation de tâche très utilisées sont les approches basées marché (ABE) [39]. Les ABE ont reçu une attention considérable [7, 15]. Ces approches consistent à reproduire au sein de l'équipe SMR, des tours d'enchères s'apparentant aux enchères que l'on peut observer dans un système économique classique, et permettent une résolution très rapide du problème d'allocation de tâches. Ces approches ont été améliorées et adaptées au cours des années pour prendre en compte de nombreux aspects spécifiques à différentes applications des systèmes multi-robots, tels des défauts dans la réception et l'émission des messages de coordination [21, 41], des pannes de robots [12, 16], ou encore les limitations des batteries des robots [5]. Pour le cadre spécifique des missions de patrouille, différentes améliorations ont été présentées, telles que des robots capables de rejoindre ou de quitter l'équipe à tout moment [33], ou l'inclusion d'un superviseur robotique redistribuant les tâches des robots peu performants [32].

Cependant, les ABE dépendent fortement de la fiabilité des communications, car la qualité de l'allocation obtenue fluctue fortement en fonction des messages d'enchère qui sont perdus. En général, les performances obtenues par les ABE se détériorent à mesure que le RC devient moins fiable [14]. Otte *et al.* [29] ont étudié l'impact de la dégradation du RC sur les performances de nombreuses variantes d'ABE. Plus précisément, ils ont modélisé la probabilité qu'un message ne parvienne pas à son destinataire grâce à deux modèles stochastiques de communication : le modèle de Bernoulli, et le modèle de Gilbert-Elliot. Ils ont mené de larges campagnes de simulations numériques et d'expérimentations en conditions réelles, pour comparer les performances de nombreuses variantes d'ABE selon la probabilité de perte des messages. En particulier, les auteurs ont considéré deux métriques de performance, qui doivent toutes deux être minimisées : les distances parcourues par les robots et le nombre de tâches allouées par défaut au commissaire priseur des enchères. Cette dernière métrique leur permet en particulier d'évaluer le nombre de tâches pour lesquelles aucune enchère n'a été reçue par le commissaire priseur, car les messages ont tous été perdus. Deux scénarios décrivant des missions multi-robots différentes ont été considérés par les auteurs. Dans le cadre du premier scénario, les robots doivent maximiser une fonction de récompense,

qui augmente lorsqu'ils exécutent des tâches dont la valeur est déterminée aléatoirement au début de la mission. Il n'y a donc pas de synergie entre les tâches. Dans le second scénario, les robots doivent visiter un ensemble de positions pour compléter la mission. Ici, les tâches ont donc des synergies, en fonction de leurs positions dans l'espace. Les résultats obtenus par les auteurs montrent que dans le scénario où les récompenses associées aux tâches sont aléatoires, la variante d'ABE la plus performante est une approche qu'ils ont conçue, tandis que dans le scénario de visite, où les tâches sont interdépendantes, l'approche par enchères Séquentielles Single Item (SSI) est la mieux adaptée. Toutefois, dans les deux scénarios, ils constatent que les performances de toutes les ABE se détériorent de façon dramatique à mesure que la probabilité de perte de messages augmente. Des communications peu fiables sont donc très préjudiciables à la qualité des solutions produites par les ABE pour l'ATMR [14, 29], ce qui motive notre étude.

De nombreuses études proposent des améliorations des protocoles de communication entre les robots. Ferri *et al.* [9] ont proposé une méthode pour réduire les exigences de communication en utilisant un système d'enchères single item (ESI). Les commissaires-priseurs n'envoient les messages d'annonce qu'aux enchérisseurs avec lesquels ils partagent immédiatement un lien de communication dans le réseau. La méthode proposée par Madhyastha *et al.* [23], basée sur un système ESI, permet aux robots de recueillir et de filtrer les offres des agents voisins. Mezei *et al.* [25] permettent aux robots recevant une annonce de décider s'ils doivent la relayer à leurs coéquipiers en fonction d'une estimation de leurs offres.

Une alternative pour assurer la stabilité du RC est de concevoir une fonction d'évaluation des mises prenant en compte la qualité du RC, afin d'inciter les robots à privilégier les tâches qui leur permettraient de maintenir un RC bien connecté. Sheng *et al.* [38] ont défini une mesure appelée mesure de proximité, dont le but est d'évaluer la proximité d'un robot par rapport à ses voisins. Cette mesure est utilisée dans la formule d'évaluation des mises pour tenir compte des modifications apportées au RC pendant le processus d'ATMR. Des simulations numériques ont permis d'établir que l'inclusion de la mesure de proximité dans l'évaluation des mises améliore la qualité des solutions produites. Cependant, cette méthode repose sur l'hypothèse forte selon laquelle la capacité des robots à établir des liens de communication ne repose que sur la distance qui les sépare. Or, dans des environnements de mission réels, des obstacles physiques peuvent empêcher deux robots de communiquer, même s'ils sont proches. Il faut donc prendre en compte de tels obstacles pour évaluer la qualité des communications en conditions réelles.

Notre étude répond à cette nécessité en introduisant notre terme de préservation de la connectivité de RC, qui dépend du nombre de robots dans les sous-RC, plutôt que de la distance entre les robots. En outre, nous abordons un scénario différent, la mission de surveillance, et considérons la possibilité que les robots rencontrent des événements dynamiques, afin d'évaluer l'effet de la prise en compte de la connectivité du RC sur la robustesse de notre approche face aux événements dynamiques.

### 3. MODÉLISATION DU PROBLÈME

#### 3.1. MISSION DE SURVEILLANCE MULTI-ROBOTS

Dans le cadre de ce papier, nous considérons la résolution du problème d'ATMR correspondant à une mission de surveillance, remplie par une équipe de robots notée  $\mathcal{R}$ . Ces robots sont capables de communication via un RC multi-hop, et sont hétérogènes dans leurs caractéristiques techniques, telles que leur vitesse de déplacement et leurs capacités de surveillance. On suppose également qu'ils sont capables de déterminer exactement leur position. La mission prend place dans un environnement connu représenté par un graphe  $\mathcal{G} = \{\mathcal{W}, \mathcal{E}\}$ . Les robots doivent surveiller les nœuds  $\mathcal{W}$  aussi souvent que possible, et ne peuvent se déplacer d'un nœud à un autre qu'en suivant les arêtes  $\mathcal{E}$  de  $\mathcal{G}$ .

Pour évaluer la qualité d'une allocation des nœuds aux robots, nous utilisons comme métrique, l'oisiveté, qui correspond au délai s'écoulant entre deux visites successives d'un même nœud. L'oisiveté associée au nœud  $w \in \mathcal{W}$  est définie par l'équation (3.1).

$$I_w(t) = t - t_{\text{last}}^w \quad (3.1)$$

Avec  $t \in \mathbb{R}_+$  la date présente au moment du calcul, et  $t_{\text{last}}^w \leq t$  la date de la dernière visite du nœud  $w$ . Dans la suite de ce papier,  $\mathbf{t}_{\text{last}}$  désigne le vecteur des dates de dernières visites  $t_{\text{last}}^w$ ,  $w \in \mathcal{W}$ . L'objectif de la mission est de minimiser l'oisiveté maximale parmi tout les nœuds, décrite par l'équation (3.2), tout au long de la mission. Par simplicité, nous appellerons cette métrique « oisiveté maximale » dans la suite de ce papier. Des modélisations similaires ont déjà été utilisées pour la résolution du problème d'ATMR dans des missions de surveillance multi-robots [22, 37, 46].

$$I^{\max}(t) = \max_{w \in \mathcal{W}} I_w(t), \quad \forall t \in [0, t_e] \quad (3.2)$$

Avec  $t_e$  la date de fin de la mission. Notons que toute allocation qui laisserait un nœud sans surveillance résulte en la pire valeur possible de l'oisiveté maximale tout au long de la mission, c'est à dire  $I^{\max}(t) = t$ ,  $\forall t \in [0, t_e]$ . En conséquence, la résolution du problème d'ATMR avec comme objectif la minimisation de  $I^{\max}(t)$  à tout instant  $t$ , favorise une répartition des robots sur l'ensemble des nœuds.

Au sein de l'équipe multi-robots, chaque robot est décrit par son type, noté  $T \in \mathcal{T}$ . Les robots du même type ont la même vitesse de déplacement, notée  $v_T$ , la même portée de communication, notée  $c_T$ , et les mêmes capacités de déplacement. Pour chaque type  $T \in \mathcal{T}$ , les capacités de déplacement sont représentées par un sous-ensemble d'arêtes  $\mathcal{E}_T \subset \mathcal{E}$ , représentant les chemins accessibles aux robots de type  $T$ . En d'autres termes, les robots de type  $T$  ne peuvent se déplacer que sur les arêtes  $\mathcal{E}_T$ . Enfin, le type des robots indique également les capteurs dont ils sont équipés. Plus précisément, étant donné un type  $T$ , on note  $\mathcal{S}_T$  l'ensemble des capteurs équipant les robots de type  $T$ . D'un autre côté, les nœuds nécessitent des capteurs spécifiques pour être surveillés, et on note l'ensemble des capteurs en mesure de surveiller le nœud  $w \in \mathcal{W}$ ,  $\mathcal{S}_w \subset \mathcal{S}$ . Étant données un nœud  $w \in \mathcal{W}$  et un robot  $r \in \mathcal{R}$  de type  $T \in \mathcal{T}$ ,  $r$  peut surveiller  $w$  si et seulement si il est équipé d'au moins un capteur adéquat, c'est à dire  $\mathcal{S}_T \cap \mathcal{S}_w \neq \emptyset$ . Cependant, les robots ne disposent pas de connaissances globales sur les ensembles

de capteurs  $S_w$ ,  $w \in \mathcal{W}$ . Ils doivent donc maintenir à jour leurs connaissances locales de ces ensembles, et on note  $S_w^r$  l'ensemble des capteurs que le robot  $r$  croit capables de surveiller le nœud  $w$ .

Les ABE sont des algorithmes de décision distribués, et les robots ont donc besoin d'échanger des informations avec leurs équipiers via le RC multi-hop, afin de maintenir à jour leur connaissance de l'environnement, de la mission et du comportement de leurs équipiers. En particulier, dans le cadre des ABE, les robots doivent partager les messages d'annonce qui débutent les enchères à autant d'équipiers que possible, afin que les tours d'enchères résultent en l'allocation la plus performante possible [29]. De plus, l'allocation courante, et les dates de dernières visites des nœuds, sont également des données essentielles pour le bon calcul des enchères dans une mission de surveillance multi-robots. Elles doivent donc être partagées aussi largement que possible à chaque modification pour permettre aux robots d'évaluer précisément leurs enchères en fonction de ces paramètres [46]. C'est pour ces raisons, que nous souhaitons maintenir la connectivité du RC qui connecte les robots.

### 3.2. ÉVÈNEMENTS DYNAMIQUES

En situation réelle, les robots sont susceptibles, pendant l'exécution de leurs tâches, de rencontrer des événements imprévus. De part leur imprévisibilité, ces événements ne peuvent pas être pris en compte lors de la résolution du problème d'ATMR. Dans le cadre de cette étude, nous nous sommes concentrés sur des événements altérant les conditions d'observation au niveau des nœuds. Par exemple, si l'éclairage artificiel d'une pièce sans fenêtre est coupé, la surveillance de cette pièce ne pourra plus s'effectuer à l'aide d'une caméra en lumière visible, et nécessitera un capteur adapté, tel qu'une caméra à vision nocturne. De la même manière, si la visibilité au niveau d'un nœud est limité par de la fumée, la surveillance de ce nœud nécessitera l'emploi d'une caméra thermique.

Un événement dynamique est représenté par le triplet  $(w \in \mathcal{W}, t > 0, \mathbf{s}^- \in \mathcal{S}_w)$ , où  $w$  indique la position de l'évènement,  $t$  sa date d'occurrence, et  $\mathbf{s}^-$  l'ensemble des capteurs qui deviennent inopérant pour surveiller  $w$ . En d'autres termes, après la date  $t$  et jusqu'à la fin de la mission, la liste des capteurs permettant de surveiller  $w$  est mise à jour de manière à ce que  $\mathcal{S}_w \leftarrow \mathcal{S}_w \setminus \mathbf{s}^-$ . Notons que les robots ne sont capables de détecter de tels événements que lorsqu'ils tentent de surveiller le nœud affecté.

## 4. PROTOCOLE D'ENCHÈRES

Pour résoudre le problème défini dans la section 3.1, nous utilisons une ABE, spécifiquement une approche d'enchères Séquentielles Single-Item (SSI) [40]. Le principal avantage des enchères SSI est qu'elles permettent de prendre en compte les synergies entre les tâches de la mission, tout en conservant l'aspect temps réel des ABE. Dans le cadre des enchères SSI, les robots enchérissent sur les tâches les unes après les autres, une tâche à la fois, dans des tours d'enchères consécutifs. A chaque nouveau tour, une nouvelle tâche est mise aux enchères, et les robots prennent en compte les

allocations déjà réalisées pour calculer la valeur de leurs mises. Le commissaire-priseur du tour d'enchère clôture le tour après un délai donné. Cela permet de laisser le temps aux robots de calculer leurs mises, tout en assurant que le processus ne reste pas bloqué dans l'attente d'une réponse d'un ou plusieurs robots dont les communications sont indisponibles.

À l'initialisation de la mission, un robot assume le rôle de commissaire-priseur et met, séquentiellement, tous les nœuds aux enchères. Ainsi, chaque nœud est le sujet d'un tour d'enchères, au cours duquel les robots estiment la mise qu'ils souhaitent émettre pour le nœud en délibération, et au terme duquel le nœud est alloué à un unique robot. Une perturbation aléatoire négligeable est appliquée aux mises pour départager les égalités. Après avoir reçu un nœud à visiter, un robot commence immédiatement à se déplacer vers lui, et informe ses coéquipiers du chemin qu'il prévoit d'emprunter. Simultanément à ses déplacements, il continue à participer aux enchères. Enfin, après avoir atteint et surveillé un nœud lui ayant été alloué, il informe ses coéquipiers du succès ou de l'échec de sa visite, puis insère ce nœud dans une queue pour le remettre aux enchères dès que possible. Ces comportements sont implémentés par les procédures décrites dans la suite de cette section.

Notons que dans le cadre des expériences de simulations décrites dans la section 5, l'unicité du rôle de commissaire-priseur est assurée par un protocole centralisé consistant à attribuer un jeton unique au premier robot l'ayant demandé. Le déploiement en conditions réelles du protocole SSI décrit dans la suite de cette section nécessite la décentralisation de ce processus, par exemple grâce à un protocole de consensus. Dans le cadre d'un tel protocole, les robots pourront échanger des messages contenant l'utilité qu'ils accordent au rôle de commissaire-priseur, qui peut être estimée en fonction de l'oisiveté des nœuds qu'ils souhaitent mettre aux enchères. Lorsqu'un robot souhaitant devenir commissaire-priseur reçoit un message contenant une valeur supérieure à sa propre utilité, il abandonne sa requête pour l'obtention du jeton. Ainsi, à la fin du consensus, les robots d'une même sous-équipe auront tous abandonné le jeton, hormis le robot associant au rôle de commissaire-priseur l'utilité la plus élevée.

Dans la suite de ce papier, on note  $A_r$  la liste des tâches à mettre aux enchères possédées par le robot  $r$ , et  $exec_r$  la liste des tâches qu'il doit exécuter. La procédure suivie par les robots lorsqu'ils atteignent un nœud contenu dans leur liste de tâches à exécuter est décrite par l'Algorithme 1. Lorsqu'un robot  $r$  arrive au niveau d'un nœud  $w \in exec_r$ , il évalue les capteurs nécessaires pour surveiller  $w$ , pour détecter si un évènement dynamique a modifié  $S_w$  (ligne 1). Si c'est le cas,  $r$  met à jour ses connaissances sur les capteurs adéquats pour surveiller  $w$  (ligne 3). Notons qu'à partir de ce moment,  $S'_w$  représente fidèlement  $S_w$  à l'instant  $t$ , dans tout les cas de figure. Ensuite,  $r$  partage cette information à ses coéquipiers via le RC multi-hop (ligne 4). S'il est équipé des capteurs adéquats pour la surveillance de  $w$ , c'est à dire si  $S_T \cap S_w \neq \emptyset$ , la date de dernière visite de  $w$  est mise à jour à la date courante,  $t$ , ce qui revient à remettre à zéro l'oisiveté de  $w$  (ligne 6). Cette information est alors partagée avec les autres robots via le RC multi-hop (ligne 7). Sinon,  $r$  ne peut pas surveiller  $w$ , et  $t'_{last}[w]$  n'est pas mise à jour, ce qui veut dire que son oisiveté continue d'augmenter. Notons

que dans ce cas de figure,  $r$  ayant mis à jour  $S_w^r$ , il pourra utiliser cette donnée pour éviter d'émettre une mise pour  $w$  à l'avenir. On assure ainsi que l'allocation de  $r$  à  $w$  ne se reproduira pas. Pour que le SMR procède constamment à des réallocations, les robots doivent remettre aux enchères chaque nœud qu'ils surveillent. Pour parvenir à ce comportement, lorsqu'un robot  $r$  atteint un nœud  $w \in exec_r$ ,  $w$  est supprimé de  $exec_r$  et inséré à la fin de  $A_r$  (lignes 8 et 9).

---

**Algorithme 1:** Comportement du robot  $r \in \mathcal{R}$  de type  $T \in \mathcal{T}$  pour la surveillance du nœud  $w \in \mathcal{W}$ .

---

**Données:** Le nœud surveillé,  $w$ ,  
 Les capteurs dont  $r$  est équipé,  $S_T$   
 Les capteurs que  $r$  croit aptes à la surveillance  $w$ ,  $S_w^r$ ,  
 La liste d'exécution de  $r$ ,  $exec_r$ ,  
 La liste des tâches à mettre aux enchères de  $r$ ,  $A_r$ ,  
 Le vecteur des dates de dernières visites selon  $r$ ,  $\mathbf{t}_{last}^r$ .

- 1  $S_w^{r'} \leftarrow \text{ÉvaluerCapteursAdéquats}(w)$ ;
- 2 **if**  $S_w^{r'} \neq S_w^r$  **then**
- 3      $S_w^r \leftarrow S_w^{r'}$ ;
- 4     **DiffuserCapteursAdéquats** ( $w, S_w^r$ );
- 5 **if**  $S_T \cap S_w^r \neq \emptyset$  **then**
- 6      $\mathbf{t}_{last}^r[w] \leftarrow t$ ;
- 7     **DiffuserDatesVisites** ( $\mathbf{t}_{last}^r$ );
- 8  $exec_r \leftarrow exec_r \setminus \{w\}$ ;
- 9  $A_r \leftarrow A_r \cup \{w\}$ ;

---

Un tour d'enchères est initié par un robot  $r$  dont la liste des tâches à mettre aux enchères,  $A_r$ , est non vide. Ce robot joue alors le rôle de commissaire-priseur pour la durée du tour d'enchère. La procédure suivie par le commissaire-priseur est décrite par l'Algorithme 2. Pour débiter une enchère, le commissaire-priseur émet un message d'annonce, contenant l'emplacement de la tâche et la date limite de l'enchère (ligne 5). Le commissaire-priseur attend ensuite de recevoir des enchères en réponse à son annonce, jusqu'à ce que la date limite de l'enchère soit atteinte (ligne 6). Lorsqu'il reçoit un message d'enchère, contenant la mise d'un robot pour la tâche annoncée, ainsi que l'identifiant de ce robot, le commissaire-priseur l'enregistre dans la liste  $\mathbf{B}_w$  (ligne 8). Le commissaire-priseur mise également pour la tâche qu'il met aux enchères, à condition de disposer des capteurs adéquats, selon son estimation de  $S_w$  (lignes 9 et 10). Ayant rassemblé autant d'enchères que possible, le commissaire-priseur peut déterminer le gagnant de l'enchère, c'est-à-dire le robot ayant déposé la mise la plus élevée (ligne 12). Il émet vers tous les enchérisseurs un message d'attribution contenant l'identifiant du robot ayant remporté l'enchère, pour que cette information soit propagée dans tout le réseau (ligne 13). Si le commissaire-priseur a lui-même déposé la meilleure mise, il remporte l'enchère, et planifie le nœud qu'il a remporté (ligne 15). Toutefois, il est possible qu'aucun robot possédant des capteurs aptes à surveiller le nœud en délibération n'ait répondu au message d'annonce envoyé par le commissaire-priseur, et

que le commissaire-priseur lui-même ne soit pas équipé de tels capteurs. Dans ce cas,  $\mathbf{B}_w = \emptyset$ , et le nœud ne peut pas être assigné. Il est alors réintroduit à la fin de la liste de tâches à enchérir du commissaire-priseur, pour être remis aux enchères plus tard (ligne 17).

---

**Algorithme 2:** Comportement du robot  $r \in \mathcal{R}$  lorsqu'il endosse le rôle du commissaire-priseur

---

**Données:** La liste des tâches à mettre aux enchères de  $r$ ,  $A_r$ ,  
 Les capteurs que  $r$  croit aptes à la surveillance  $w$ ,  $S_w^r$ ,  
 Les capteurs dont  $r$  est équipé,  $S_T$ ,  
 La durée  $\bar{t}$  d'un tour d'enchères.

```

1  $w \leftarrow A_r[0]$ ;
2  $A_r \leftarrow A_r \setminus \{w\}$ ;
3  $t_0 \leftarrow t$ ;
4  $\mathbf{B}_w \leftarrow \emptyset$ ;
5 DiffuserAnnonce ( $w, t_0 + \bar{t}$ );
6 while  $t \leq t_0 + \bar{t}$  do
7   | if RecevoirMise ( $B_{r'}(w)$ ) then
8   |   |  $\mathbf{B}_w \leftarrow \mathbf{B}_w \cup B_{r'}(w)$ ;
9   | if  $S_T \cap S_w^r \neq \emptyset$  then
10  |   |  $\mathbf{B}_w \leftarrow \mathbf{B}_w \cup \text{CalculerMise}(w)$ ;
11 if  $\mathbf{B}_w \neq \emptyset$  then
12  |   |  $r^* \leftarrow \arg \max \mathbf{B}_w$ ;
13  |   | DiffuserRésultat ( $r^*, w$ );
14  |   | if  $r^* = r$  then
15  |   |   | Planifier ( $w$ );
16 else
17 |   |  $A_r \leftarrow A_r \cup \{w\}$ ;

```

---

Lorsqu'un robot ne joue pas le rôle du commissaire-priseur, il est enchérisseur, et son comportement est décrit par l'Algorithme 3. Les robots enchérisseurs attendent de recevoir un message d'annonce de la part d'un commissaire-priseur (ligne 1). À la réception d'un message d'annonce, s'il dispose de capteurs aptes à la surveillance du nœud mis aux enchères, l'enchérisseur émet un message contenant son identifiant et sa mise, pour participer à l'enchère (lignes 2 et 3). D'un autre côté, si un robot enchérisseur reçoit un message de résultat indiquant qu'il a remporté une enchère, il planifie le nœud correspondant (lignes 4 à 6).

**PROPOSITION 4.1.** — *Soit  $[t_d; t_f] \subset \mathbb{R}_+^2$ ,  $t_d \leq t_f$  une période au cours de laquelle la mission a lieu, et  $\bar{t}$  la durée d'un tour d'enchères. Le nombre de messages échangés dans le cadre du protocole SSI décrit précédemment est inférieur à  $\frac{t_f - t_d}{\bar{t}} (|\mathcal{R}|^2 - 1)$ .*

*Démonstration.* — Dans le pire des cas, les tours d'enchères se succèdent sans interruption, avec une fréquence d'un tour toutes les  $\bar{t}$  secondes. Le nombre maximal

---

**Algorithme 3:** Comportement du robot  $r \in \mathcal{R}$  lorsqu'il est enchérisseur

---

**Données:** Les capteurs que  $r$  croit aptes à la surveillance  $w$ ,  $S_w^r$ ,  
 Les capteurs dont  $r$  est équipé,  $S_T$ .

```

1 if RecevoirAnnonce ( $w, \bar{t}$ ) then
2   if  $t < \bar{t}$  et  $S_T \cap S_w^r \neq \emptyset$  then
3     DiffuserMise ( $r$ , CalculerMise ( $w$ ));
4     if RecevoirRésultat ( $r^*, w$ ) then
5       if  $r^* = r$  then
6         Planifier ( $w$ );
    
```

---

de tours d'enchères est donc  $\frac{t_f - t_d}{\bar{t}}$ . Chaque tour comprend 1 message d'annonce et au plus 1 message de résultat, ainsi qu'au plus  $|\mathcal{R}| - 1$  messages de mises, dans le cas où tout les robots, sauf le commissaire-priseur, émettent un message de mise. À ce niveau, on a donc  $1 + |\mathcal{R}|$  messages par tour d'enchères. Il faut également prendre en compte que les robots répètent les messages qu'ils reçoivent de leur coéquipiers, pour assurer la communication multi-hop. Si le RC est connexe, chaque message est répété au plus  $|\mathcal{R}| - 1$ , c'est à dire par tout les robots, excepté l'émetteur initial du message. Le nombre total de messages dans le pire des cas est donc donné par

$$\frac{t_f - t_d}{\bar{t}} \cdot (1 + |\mathcal{R}|) \cdot (|\mathcal{R}| - 1) = \frac{t_f - t_d}{\bar{t}} \cdot (|\mathcal{R}|^2 - 1) \quad \square$$

Parmi les opérations décrites dans les Algorithmes 2 et 3, l'évaluation des mises (fonction CalculerMise) est particulièrement importante. En effet, c'est cette évaluation qui permet aux robots d'exprimer leurs préférences au commissaire-priseur, et au commissaire-priseur de choisir l'enchérisseur remportant le tour d'enchère. Dans la suite de cette section, nous introduisons une fonction d'évaluation des mises permettant d'estimer l'impact d'une allocation sur la connectivité du RC, en anticipant ses évolutions futures. Nous décrivons également deux fonctions d'évaluation des mises issues de la littérature.

#### 4.1. OISIVETÉ ESPÉRÉE [46]

Lorsque le robot  $r$  doit calculer sa mise pour le nœud  $w$ , le terme d'oisiveté « espérée » proposé par Yan et Zhang [46] permet d'estimer l'oisiveté de  $w$  au moment où  $r$  pourra le surveiller. Pour ce faire, le terme d'oisiveté « espérée » compare l'oisiveté instantanée de  $w$  connue par  $r$  avec le temps de trajet dont  $r$  a besoin pour se rendre en  $w$ . Nous avons adapté ce terme à notre protocole SSI, dans lequel les nœuds ne peuvent être alloués qu'à au plus un robot, pour obtenir le terme décrit par l'équation (4.1).

$$J_r(t, w) = 1 + \frac{t - \mathbf{t}_{\text{last}}^r(w)}{d(r, w)/v_T} \quad (4.1)$$

Avec  $t \in \mathbb{R}_+$ , l'instant auquel le calcul de la mise a lieu,  $\mathbf{t}_{\text{last}}^r(w) \leq t$  la date de la dernière visite de  $w$  selon les connaissances locales de  $r$ ,  $T \in \mathcal{T}$  le type de robot auquel

$r$  appartient, et  $d(r, w) \in \mathbb{R}_+$  la longueur du plus court chemin entre le dernière nœud du plan de  $r$  et  $w$ .

Toutefois, le terme  $J_r$  tel qu'il est défini par l'équation (4.1) peut croître arbitrairement à mesure que  $t$  augmente, et, en pratique, prend des valeurs beaucoup plus grandes que les termes décrits par les équations (4.4) et (4.6) que nous verrons plus tard dans cette section, ce qui rend ces termes négligeables devant le terme d'oisiveté. En conséquence, nous appliquons à  $J_r$  la transformation décrite par l'équation (4.2), pour obtenir le terme  $I_r$ , dont les valeurs évoluent entre 0,5 and 1.

$$I_r(t, w) = \frac{J_r(t, w)}{1 + J_r(t, w)} \quad (4.2)$$

PROPOSITION 4.2. — *La complexité en temps du calcul de  $I_r$  est :*

$$O(|\mathcal{E}| + |\mathcal{W}| \cdot \log |\mathcal{W}|) \quad (4.3)$$

*Démonstration.* — Tout les termes utilisés dans le calcul de  $I_r$  sont des données accessibles en  $O(1)$ , sauf  $d(r, w)$ , qui requiert de trouver le plus court chemin entre deux nœuds du graphe. On utilise l'algorithme de Dijkstra [8] pour calculer ce plus court chemin, dont la complexité est  $O(|\mathcal{E}| + |\mathcal{W}| \cdot \log |\mathcal{W}|)$  [10]. Remarquons cependant que dans l'hypothèse où  $\mathcal{G}$  est statique tout au long de la mission, l'ensemble des plus courts chemins peut être calculé en pré-processing, puis accéder en  $O(1)$  pendant les tours d'enchères.  $\square$

#### 4.2. MESURE DE PROXIMITÉ [38]

Dans ce paragraphe, nous décrivons la mesure de proximité telle que définie par Sheng *et al.* [38]. Cette mesure permet d'évaluer l'impact des allocations sur l'évolution du RC, en anticipant la distance qui sépare les robots lorsque l'allocation sera réalisée. Son objectif est de maintenir au moins un voisin à portée de chaque robot, tout en n'incitant pas les robots à s'agglomérer de façon compacte, et ainsi, d'obtenir un réseau de communication connecté tout en maintenant une répartition spatiale équilibrée. La mesure de proximité est décrite par l'équation (4.4).

$$S_r(t, w, \mathbf{d}) = e^{-\frac{d_1}{cT}} + \lambda e^{-\frac{d_2}{cT}} + \dots + \lambda^{n_k-2} e^{-\frac{d_{n_k}}{cT}} \quad (4.4)$$

Avec  $\mathbf{d} = \{d_1, \dots, d_{n_k}\}$  les distances deux à deux entre le robot enchérisseur et les autres robots évoluant au sein du même sous-RC, triées par ordre croissant.  $\lambda$  est un facteur inférieur à 1, dont l'utilité est de permettre que chaque robot de  $\mathbf{d}$  ait un impact exponentiellement plus petit sur la valeur de  $S_r$  par rapport au robot qui le précède. Ce paramètre permet de ne pas inciter les robots à former des groupes très compacts.

PROPOSITION 4.3. — *La complexité en temps du calcul de  $S_r$  est :*

$$O(|\mathcal{E}| + (|\mathcal{W}| + |\mathcal{R}|) \cdot \log |\mathcal{W}| + |\mathcal{R}| \cdot \log |\mathcal{R}|) \quad (4.5)$$

*Démonstration.* — Le calcul de  $S_r$  est composé de trois algorithmes s'exécutant séquentiellement :

- (1) Un algorithme de Dijkstra pour estimer la date d'arrivée de  $r$  sur le nœud en délibération, afin de déterminer la date à laquelle il faut anticiper les positions des autres robots. La complexité de cette étape est  $O(|\mathcal{E}| + |\mathcal{W}| \log |\mathcal{W}|)$  [10],
- (2) L'estimation de la position des autres robots du sous-RC auquel  $r$  appartient. Dans le pire des cas, ce sous-RC est composé de  $|\mathcal{R}| - 1$  robots, et le nombre maximum de nœuds dans le chemin prévu d'un robot est  $|\mathcal{W}|$ . L'estimation de la position d'un robot à la date  $t$  étant donné son chemin prévu, s'exécute en  $O(\log |\mathcal{W}|)$  car il s'agit de trouver le nœud du chemin prévu atteint à la date la plus proche de la date à laquelle  $r$  atteint le nœud en délibération. Le chemin prévu étant ordonné par date de visite, on peut utiliser une bissection. L'estimation de la position des robots s'exécute donc en  $O(|\mathcal{R}| \cdot \log |\mathcal{W}|)$ ,
- (3) Comme indiqué précédemment, il faut enfin trier la liste des distances deux à deux entre les positions estimées et la position du nœud en délibération. Le nombre maximal d'éléments de cette liste étant  $|\mathcal{R}|$ , ce tri s'exécute en  $O(|\mathcal{R}| \cdot \log |\mathcal{R}|)$ .

Ces trois procédures s'exécutant séquentiellement, la complexité totale du calcul de  $S_r$  est la somme de leurs complexités.  $\square$

#### 4.3. RÉCOMPENSER LES ALLOCATIONS AMÉLIORANT LA CONNECTIVITÉ DU RC

Pour préserver un RC aussi connecté que possible tout au long de la mission du SMR, nous proposons une fonction d'évaluation des mises dédiée à cet aspect. Ce terme, noté  $K_r(t, w)$ , est décrit par l'équation (4.6).

$$K_r(t, w) = \frac{\left| C_{\mathcal{N}\left(t + \frac{d(r,w)}{v_r}\right)}(r) \right|}{\left| C_{\mathcal{N}(t)}(r) \right|} - 1 \quad (4.6)$$

Où  $\mathcal{N}(t) = (\mathcal{R}, \mathcal{F})$  décrit le RC à l'instant  $t$ , et  $C_{\Gamma}(r)$  est la composante connexe du graphe  $\Gamma$  contenant  $r$ . Notons que chaque robot garde une version locale de  $\mathcal{N}(t)$ , de sorte que le calcul de  $K_r(t, w)$  ne prend en compte que les informations disponibles pour  $r$ .

Le dénominateur de  $K_r$  correspond au nombre de robots présents dans le même sous-RC que  $r$  au début de l'enchère. Formellement, le nombre de robots présents dans le même sous-RC que  $r$  correspond au cardinal de la composante connexe du RC qui contient  $r$ . Le numérateur de  $K_r$  est l'estimation du nombre de robots présents dans le même sous-RC que  $r$  à l'instant futur  $t + \frac{d(r,w)}{v_r}$  où  $r$  atteindra le nœud  $w$ . Pour calculer ce numérateur, il faut donc anticiper les positions des coéquipiers de  $r$ , ce qui suppose que l'on dispose d'une bonne connaissance de leurs destinations au niveau local. Enfin, notons que l'on soustrait 1 au ratio pour s'assurer que  $K_r(t, w)$  est négatif si l'allocation de  $w$  à  $r$  diminue le nombre de robots dans son sous-RC.

PROPOSITION 4.4. — *La complexité en temps du calcul de  $K_r$  est :*

$$O(|\mathcal{E}| + (|\mathcal{W}| + |\mathcal{R}|) \cdot \log |\mathcal{W}| + |\mathcal{R}|^2) \quad (4.7)$$

*Démonstration.* — Le calcul de  $K_r$  est composé de trois algorithmes s'exécutant séquentiellement :

- (1) Un algorithme de Dijkstra pour estimer la date d'arrivée de  $r$  sur le nœud en délibération, afin de déterminer la date à laquelle il faut anticiper les positions des autres robots. La complexité de cette étape est  $O(|\mathcal{E}| + |\mathcal{W}| \cdot \log |\mathcal{W}|)$  [10],
- (2) L'estimation de la position des autres robots du sous-RC auquel  $r$  appartient. Nous avons déjà montré dans la Preuve 4.2 que l'estimation de la position des robots s'exécute en  $O(|\mathcal{R}| \cdot \log |\mathcal{W}|)$ ,
- (3) Le calcul du nombre de robots dans le même sous-RC que  $r$ . Ce nombre correspondant au nombre de nœuds dans la composante connexe du RC contenant  $r$ , il suffit de déterminer les composantes connexes du RC pour l'obtenir. Pour cela, on exécute un parcours en profondeur sur le RC. Dans le pire des cas, le RC est un graphe complet à  $|\mathcal{R}|$  sommet, ce qui porte le temps de calcul de son parcours en profondeur à  $\Theta\left(|\mathcal{R}| + \frac{|\mathcal{R}|(|\mathcal{R}|-1)}{2}\right)$  [6].

Ces trois procédures s'exécutant séquentiellement, la complexité totale du calcul de  $S_r$  est la somme de leurs complexités.  $\square$

#### 4.4. AGRÉGATION DES FONCTIONS D'ÉVALUATION DES MISES

Dans les simulations que nous présenterons dans la suite de ce papier, nous exprimons la valeur des mises comme la combinaison de  $I_r$ , qui permet de prendre en compte l'oisiveté des nœuds, et d'un des deux termes permettant d'estimer l'impact des allocations sur le RC,  $S_r$  et  $K_r$ . La principale contribution de ce papier est de montrer qu'exprimer la valeur des mises en fonction de  $I_r$  et  $K_r$ , tel que décrit par l'équation (4.8a) permet d'obtenir des allocations plus robustes face aux événements dynamiques. Pour ce faire, nous avons comparé cette évaluation des mises avec deux évaluations concurrentes. La première, n'utilisant que  $I_r$ , ne prend pas en compte l'évolution du RC, et la seconde, composée de  $I_r$  et  $S_r$  comme décrit par l'équation (4.8b), utilise la mesure de proximité pour prendre en compte l'évolution du RC.

$$B_r(t, w) = I_r(t, w) + K_r(t, w) \quad (4.8a)$$

$$B_r(t, w, \mathbf{d}) = I_r(t, w) + S_r(t, w, \mathbf{d}) \quad (4.8b)$$

#### 4.5. DONNÉES PARTAGÉES

Pour calculer les mises qu'ils doivent émettre pour les nœuds en délibération, les robots doivent échanger des données sur l'état actuel de la mission. Les robots partagent leurs connaissances locales aussi largement que possible à travers le RC multi-hop, c'est à dire à tous les équipiers dans le même sous-RC.

Dans un premier temps, pour évaluer le terme d'oisiveté  $I_r$  décrit dans la section 4.1, les robots doivent partager leurs connaissances des dates de dernière visite de tous les nœuds, de façon à évaluer l'oisiveté de ces nœuds. Pour ce faire, chaque robot  $r \in \mathcal{R}$  maintient sa propre version du vecteur des dates de dernière visite,  $\mathbf{t}'_{\text{last}}$ . Ce vecteur

est mis à jour lorsque  $r$  reçoit le vecteur d'autres robots. En particulier, les robots partagent  $t_{\text{last}}^r$  avec leurs coéquipiers après avoir surveillé un nœud. A la réception d'un tel message, les robots mettent à jour leur version locale de  $t_{\text{last}}^r$  en gardant le maximum nœud par nœud de la version reçue et de leur version locale.

PROPOSITION 4.5. — Soit  $d^{\min}$  la longueur de l'arc le plus court de  $\mathcal{E}$ , et  $v^{\max}$  la vitesse du type de robot le plus rapide. La taille totale des messages de dates de dernières visites échangés au cours de la période  $[t_d; t_f]$  est inférieure à :

$$\frac{t_f - t_d}{d^{\min} \cdot v^{\max}} \cdot |\mathcal{R}|^2 \cdot |\mathcal{W}| \quad (4.9)$$

*Démonstration.* — Les messages de dates de dernières visites ne sont émis qu'après qu'un robot ait surveillé un nœud, comme décrit dans l'Algorithme 1. La fréquence à laquelle les robots surveillent des nœuds est borné par  $d^{\min} \cdot v^{\max}$ . Le nombre de nœuds surveillés par un unique robot pendant la période  $[t_d; t_f]$  est donc inférieur ou égal à  $\frac{t_f - t_d}{d^{\min} \cdot v^{\max}}$ . Dans le pire des cas, tout les robots se déplacent à la même vitesse  $v^{\max}$  et tout les arcs ont la même longueur  $d^{\min}$ , auquel cas le SMR procède à  $|\mathcal{R}| \cdot \frac{t_f - t_d}{d^{\min} \cdot v^{\max}}$  surveillances au cours de la mission. De plus, les messages de dernières visites sont de taille  $|\mathcal{W}|$  car ils décrivent la date de dernière visite de chaque nœud  $w \in \mathcal{W}$ . Enfin, chaque message est relayé au plus  $|\mathcal{R}| - 1$  fois, lorsque le RC est connexe. En multipliant le nombre maximal de messages émis par leur taille, et en considérant un RC restant connexe tout au long de la période  $[t_d; t_f]$ , on obtient le résultat énoncé dans l'équation (4.9).  $\square$

De plus, les robots doivent aussi connaître les chemins prévus par leurs coéquipiers, dont ils ont besoin pour anticiper les positions futures des autres robots et calculer les valeurs des termes d'évaluation de l'évolution du RC décrit dans les sections 4.2 et 4.3. Le chemin qu'un robot prévoit d'emprunter n'est modifié que lorsqu'il reçoit un nœud après avoir remporté un tour d'enchères. Il est donc suffisant, pour maintenir une version à jour de cette information, de ne la partager que lorsque qu'un robot remporte une enchère, et de ne partager que le chemin de ce robot. Notons que la taille maximale des messages de chemin est atteinte lorsqu'un unique robot surveille tout les nœuds de la mission, et est donc bornée par  $|\mathcal{W}|$ .

PROPOSITION 4.6. — La taille totale des messages de chemins prévus échangés au cours de la période  $[t_d; t_f]$  est inférieure à :

$$\frac{t_f - t_d}{\bar{t}} \cdot |\mathcal{R}| \cdot |\mathcal{W}| \quad (4.10)$$

*Démonstration.* — Les messages des chemins prévus ne sont émis qu'après qu'un robot ait remporté une enchère. On a établi dans la preuve de la proposition 4.1, que le nombre maximal de tours d'enchères lancés au cours de la période  $[t_d; t_f]$  est  $\frac{t_f - t_d}{\bar{t}}$ . Dans le pire des cas, les messages de chemins prévus sont de taille  $|\mathcal{W}|$ , si l'émetteur surveille tout les nœuds de la mission. Le nombre maximal de répétitions de ces messages est atteint lorsque le RC est connexe pendant toute la mission et est de  $|\mathcal{R}| - 1$  répétitions. En multipliant le nombre maximal de message par leur taille, et le nombre maximal de répétitions, on obtient le résultat présenté dans l'équation (4.10).  $\square$

## 5. EXPÉRIENCES DE SIMULATION

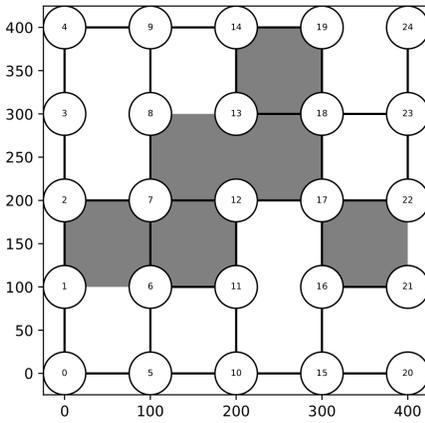
### 5.1. CONFIGURATION DES CAMPAGNES DE SIMULATIONS

L'objectif des simulations décrites dans cette section est d'évaluer l'impact de l'utilisation du terme de connectivité du RC,  $K_r$  (décrit dans la section 4.3), sur la qualité des allocations produites par le protocole SSI. Pour ce faire, nous comparerons les oisivetés maximales associées à ces allocations, avec celles associées à des allocations obtenues en utilisant le terme d'oisiveté  $I_r$  [46] décrit dans la section 4.1 et la mesure de proximité [38] décrite dans la section 4.2.

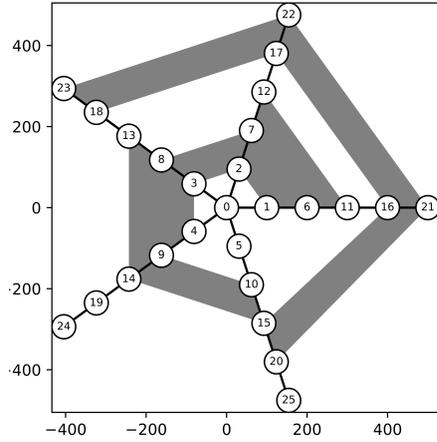
Plus précisément, nous avons évalué 3 méthodes : un protocole SSI simple n'utilisant que le terme d'oisiveté (notée  $B = I$  dans les résultats), un protocole SSI utilisant la mesure de proximité de Sheng [38] (notée  $B = I + S$ ), et le protocole SSI utilisant notre contribution, c'est à dire notre fonction d'évaluation de la connectivité du RC ayant pour but d'améliorer les communications au sein de SMR (notée  $B = I + K$ ). Pour mesurer les performances de ces approches, nous avons mesuré l'évolution de l'oisiveté maximale,  $I^{\max}$ , telle que définie par l'équation 3.2. L'objectif de toutes les méthodes est de minimiser  $I^{\max}$ .

Ces comparaisons ont été réalisées sur différents ensembles d'instances générées aléatoirement, ainsi que sur des instances inspirées de missions réelles. Des instances ont été générées aléatoirement selon deux topographies : des grilles, desquelles on retire des edges aléatoirement, et des étoiles. Dans le cadre de nos simulations, la capacité à communiquer d'une paire de robots dépend de leur portée de communication, et d'obstacles physiques qui gênent la réception des messages. En particulier, on considère que deux robots n'ayant pas de ligne de vue l'un sur l'autre ne peuvent pas communiquer. Des obstacles sont placés aléatoirement dans ces instances, de manière à ce qu'en moyenne, une cellule sur 4 abrite un obstacle. Les figures 5.1a et 5.1b représentent des exemples d'instances générées aléatoirement. Les instances inspirées de conditions réelles sont représentées par les figures 5.2a et 5.2b. Pour ces instances, les obstacles correspondent aux bâtiments et aux arbres, et ont toujours les mêmes positions. Les arrêtes étant, par définition, libres d'obstacles, elles constituent toujours une ligne de vue permettant à des robots situés à leurs extrémités de communiquer. Par exemple, dans le scénario présenté dans la figure 5.1b, les obstacles présents autour de l'arrête [2, 7] n'empêchent pas deux robots se trouvant l'un en 2, l'autre 7, d'établir un lien de communication.

De plus, nous avons généré, dans toutes les instances, des événements dynamiques, tels que décrits dans la section 3.2. Les événements sont générés aléatoirement tel qu'en moyenne, la moitié des nœuds nécessiteront une caméra à vision nocturne, et un quart des nœuds nécessiteront une caméra thermique. Ces événements surviennent entre  $t = 50$  secondes et  $t = 900$  secondes, et chaque mission dure 3 600 secondes pour que le processus d'ATMR ait le temps de converger vers une solution stable. Cela permet également d'examiner l'effet des événements dynamiques sur l'oisiveté des nœuds. La portée de communication des robots est  $c_T = 300$  pour tous les types de robots.

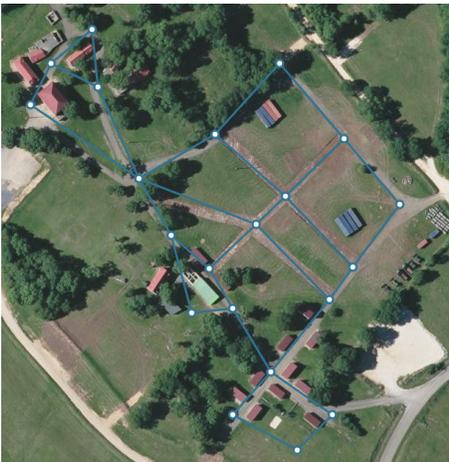


(a) Instance générée aléatoirement, de type grille  $5 \times 5$ .



(b) Instance générée aléatoirement, de type étoile avec 5 branches de profondeur 5.

FIGURE 5.1 – Exemples d’instances générées aléatoirement. Les cellules grisées représentent des obstacles qui bloquent les communications.



(a) Instance de petite taille, inspirée d’une mission réelle. Le graphe est composé de 21 nœuds.



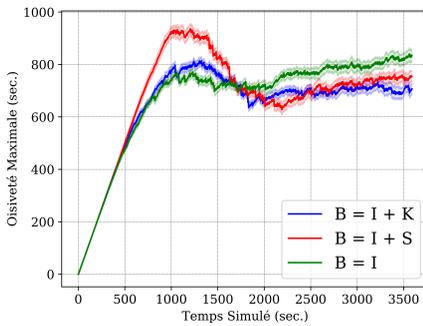
(b) Instance de grande taille, inspirée d’une mission réelle. Le graphe est composé de 45 nœuds.

FIGURE 5.2 – Vues satellites des instances inspirées de conditions réelles. Les obstacles correspondent aux bâtiments et aux arbres.

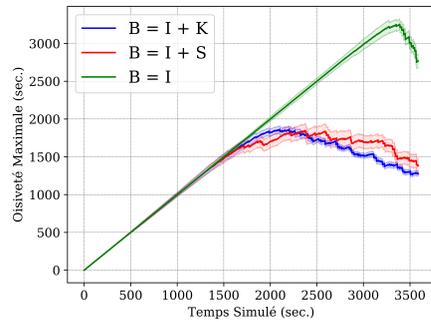
Pour obtenir les résultats décrits dans la section 5.2, nous avons conduit des missions avec une équipe de 5 robots sur 50 instances en grille  $5 \times 5$ , 50 instances en étoiles avec 5 branches de profondeur 5, et 50 instances correspondant à la mission réelle de petite taille (figure 5.2a). Nous avons également conduit des missions avec une équipe de 10 robots sur 50 instances en grille  $10 \times 10$ , 50 instances en étoiles avec 10 branches de profondeur 10, et 50 instances correspondant à la mission réelle de grande taille (figure 5.2b).

## 5.2. DISCUSSION DES RÉSULTATS

Les résultats des missions générées aléatoirement de type grille  $5 \times 5$  sont présentés dans la figure 5.3a.



(a) Missions générées aléatoirement, de type grille  $5 \times 5$ , soit 25 nœuds.



(b) Missions générées aléatoirement, de type étoile avec 5 branches de profondeur 5, soit 26 nœuds.

FIGURE 5.3 – Résultats des missions générées aléatoirement et exécutées avec une équipe de 5 robots, pour l’oisiveté maximale au long des missions. 50 instances ont été moyennées pour obtenir les résultats présentés. Les bandes de couleur autour de chacune des courbes correspondent aux intervalles de confiance avec  $\alpha = 0,05$ .

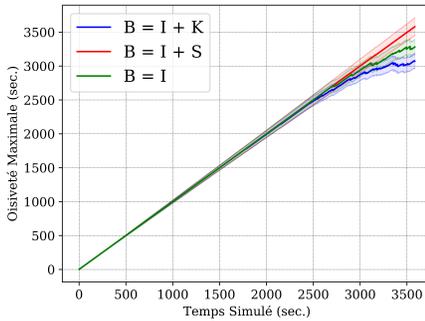
Ces missions ont été exécutées par une équipe de 5 robots. On voit que l’inclusion de  $K_r$  dans la fonction d’évaluation des mises (courbe bleue) permet au SMR de réagir plus rapidement aux évènements dynamiques qu’en incluant  $S_r$  dans la fonction d’évaluation des mises (courbe rouge). Cela montre que les communications sont améliorées par notre approche dans une plus grande proportion qu’en utilisant la mesure de proximité,  $S_r$ . Bien que l’approche ne considérant pas les communications (courbe verte) soit plus performantes en début de mission, on voit que la comparaison avec notre approche s’inverse après un certain temps. On peut expliquer ce comportement car en début de mission, cette approche n’optimise que l’oisiveté, et visite donc plus rapidement tous les nœuds. Puis, après un certain temps, le RC mieux connecté maintenu grâce à notre approche permet de procéder à de meilleures réallocations dynamiques,

qui permettent au SMR de conserver une oisiveté maximale inférieure. Ces réallocations de bonne qualité sont possibles si un plus grand nombre de robots participent à chaque enchère, et donc si le RC est mieux connecté.

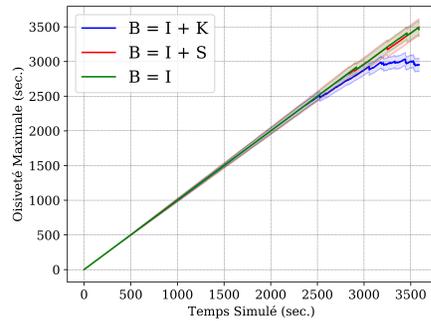
Les résultats obtenus pour les missions générées aléatoirement de type étoile avec 5 branches de profondeur 5, présentées dans la figure 5.3b, sont différents, mais notre méthode a encore produit des allocations plus performantes. Les graphes en forme d'étoile sont deux fois plus étendus que les graphes en forme de grille, ce qui rend les communications beaucoup plus rares. Ils permettent de constater l'incapacité de l'approche ne prenant pas en compte les communications (courbe verte) à réallouer des nœuds ayant subi des événements dynamiques, lorsque des liens de communications ne s'établissent pas régulièrement par hasard. Ainsi, les deux approches prenant en compte l'évolution du RC dans leur fonction d'évaluation des mises ont été largement plus performantes, car le maintien d'un RC bien connecté leur a permis de rapidement réallouer ces nœuds. Bien que l'on puisse remarquer qu'aucune des trois approches présentées dans la figure 5.3b ne semble avoir terminé de converger vers une oisiveté stable avant la fin de la mission, notre approche (courbe bleue) a plus rapidement produit des allocations induisant de faibles oisiveté maximales, car elle permet de conserver un RC mieux connecté par rapport à la méthode utilisant la mesure de proximité (courbe rouge). Comme discuté précédemment, cela permet à notre approche de faire participer plus de robots aux tours d'enchères, et d'accélérer la convergence vers des allocations plus performantes.

Les résultats obtenus sur les missions générées aléatoirement et exécutées par une équipe de 10 robots (figure 5.4) sont très similaires pour les missions de type grille  $10 \times 10$  et étoile avec 5 branches de profondeur 10. La première observation que l'on peut faire de ces résultats est que le SMR a eu beaucoup de difficultés à visiter tous les nœuds au moins une fois, même après 3 600 secondes d'exécution. On peut également observer que l'approche utilisant notre terme de maintien de la connectivité du RC,  $K_r$  (courbe bleue) est plus performante que les approches utilisant uniquement  $I_r$  (courbe verte) ou  $I_r + S_r$  (courbe rouge), puisqu'elle parvient à visiter tous les nœuds au moins une fois plus souvent.

Pour les missions de type grille  $10 \times 10$ , ces résultats s'expliquent par deux facteurs. Tout d'abord, l'augmentation du ratio entre le nombre de nœuds et le nombre de robots, qui passe de 5 pour les missions de type grille  $5 \times 5$  à 10 pour les missions de type grille  $10 \times 10$ , explique en partie l'augmentation de l'oisiveté maximale et du délai avant que tous les nœuds n'aient été visités au moins une fois. Intuitivement, le doublement de ce ratio devrait induire un doublement de l'oisiveté maximale et du délai. Le second phénomène expliquant la difficulté des missions de type grille  $10 \times 10$  est l'étalement de la zone de la mission. Chaque nœud de la grille étant placé à une distance de 100 mètres de ses voisins, l'aire de la zone des missions de type grille  $10 \times 10$  est de 1 kilomètre carré, tandis que l'aire de la zone des missions de type grille  $5 \times 5$  n'est que de 0,25 kilomètre carré. D'un autre côté, le doublement du nombre de robots dans le SMR entraîne un doublement de l'aire couverte par les portées de communication des robots. On voit donc que l'aire de la zone de mission est multipliée par 4 tandis



(a) Missions générées aléatoirement, de type grille  $10 \times 10$ , soit 100 nœuds.



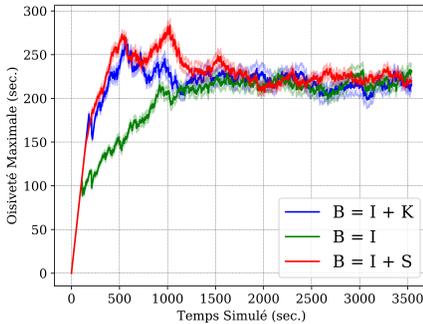
(b) Missions générées aléatoirement, de type étoile avec 5 branches de profondeur 10, soit 51 nœuds.

FIGURE 5.4 – Résultats des missions générées aléatoirement et exécutées avec une équipe de 10 robots, pour l’oisiveté maximale au long des missions. 50 instances ont été moyennées pour obtenir les résultats présentés. Les bandes de couleur autour de chacune des courbes correspondent aux intervalles de confiance avec  $\alpha = 0,05$ .

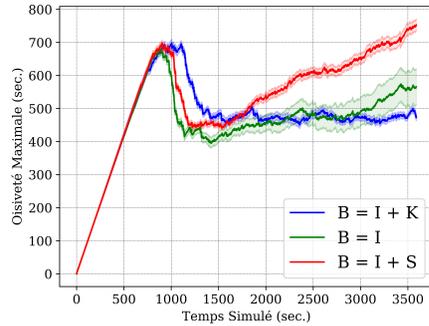
que l’aire dans laquelle les robots peuvent communiquer n’est que doublée. Cela rend les communications beaucoup plus rares, et la réallocation des nœuds ayant subi des évènements dynamiques est donc très difficile. Notre approche étant la seule à correctement évaluer l’impact des allocations sur l’évolution du RC, elle permet de résoudre ce problème dans une certaine mesure.

Ces deux explications ne sont pas valables pour les missions de type étoile avec 5 branches de profondeur 10. En effet, par rapport aux missions de type étoile avec 5 branches de profondeur 5, le ratio entre le nombre de nœuds à visiter et le nombre de robots reste le même ( $\approx 5$ ), et l’étalement de la zone ne change pas, tandis que le nombre de robots double. En revanche, le nombre d’obstacles présents dans l’environnement de la mission, qui dépend du nombre de cellules de  $\mathcal{G}$ , double. Cela dégrade les opportunités des robots pour établir des liens de communications, ce qui explique la dégradation de l’oisiveté maximale.

Les résultats sur les missions générées à partir de données réelles, présentés dans la figure 5.5 sont plus mitigés. Les résultats obtenus pour les missions générées à partir de la mission réelle de petite taille, comptant 21 nœuds et exécutées par une équipe de 5 robots, montrent les limites de l’intérêt d’inciter les robots à conserver un RC bien connecté. En effet, on voit sur la figure 5.5a que l’approche ne tenant compte que de l’oisiveté des nœuds,  $B_r = I_r$  (courbe verte), est plus performante que les deux approches incluant une évaluation de l’évolution du RC (courbes bleue et rouge). Pour expliquer ce phénomène, il faut de nouveau comparer l’aire de la zone sur laquelle les missions prennent place avec les aires couvertes par les portées de communication des robots. En effet, les nœuds sont inclus dans un rectangle de 216 mètres sur 243, tandis



(a) Missions inspirées d'une configuration réelle, de petite taille (21 nœuds), réalisées par une équipe de 5 robots.



(b) Missions inspirées d'une configuration réelle, de grande taille (46 nœuds), réalisées par une équipe de 10 robots.

FIGURE 5.5 – Résultats des missions inspirées par des missions réelles. 50 instances ont été moyennées pour obtenir les résultats présentés. Les bandes de couleur autour de chacune des courbes correspondent aux intervalles de confiance avec  $\alpha = 0,05$ .

que la portée de communication des robots est un disque est 300 mètres de rayon. Les robots sont donc toujours à portée de communication de leurs équipiers. De plus, les obstacles étant surtout répartis sur les bords de la zone de mission, ils ne scindent pas la zone en plusieurs parties, et ont donc peu d'impact. Pour ces raisons, l'inclusion des termes  $S_r$  ou  $K_r$  ne permet pas d'améliorer la connectivité du RC, mais parasitent l'optimisation de l'oisiveté, en ajoutant des biais inutiles au terme  $I_r$ . Cela explique les bonnes performances de l'approche basant le calcul des mises uniquement sur  $I_r$  (courbe verte). Toutefois, on note que notre approche produit des oisivetés inférieures à celles produites en utilisant la mesure de proximité. En effet,  $S_r$  étant basé sur les distances entre les robots, il peut impacter les décisions prises par le protocole SSI même si le RC est connexe. Ainsi, si le RC reste connexe mais que les robots s'éloignent les uns des autres,  $S_r$  prendra une valeur plus petite pour signifier que l'allocation n'est pas désirable. Au contraire,  $K_r$  est nul si le RC reste connexe et n'a donc aucun effet dans ce cas.

Pour les missions générées à partir du graphe réel de grande taille, comptant 46 nœuds et exécutées par une équipe de 10 robots (figure 5.5b), on constate que notre approche (courbe bleue) a été plus lente pour commencer sa convergence vers des allocations performantes que les deux autres approches étudiées. En effet, on voit que l'oisiveté maximale commence à diminuer autour de  $t = 900$  secondes pour l'approche ne prenant en compte que  $I_r$  pour évaluer les mises (courbe verte), et autour de  $t = 1\,000$  secondes pour l'approche utilisant  $I_r + S_r$ , alors que notre approche ne parvient à diminuer l'oisiveté maximale qu'après  $t = 1\,100$  secondes. Toutefois, une fois que notre approche a convergé vers des allocations performantes, elle parvient à les maintenir tout au long des missions. Au contraire, on voit que les courbes

représentant les approches  $B_r = I_r$  et  $B_r = I_r + S_r$  augmentent constamment à mesure que le temps simulé s'écoule. Ce comportement est dû à un nombre limité d'instances, au cours desquelles ces méthodes ne sont jamais parvenu à réallouer un ou plusieurs nœuds ayant subi des évènements dynamiques. En conséquence, l'oisiveté de ces nœuds n'a pas cessé d'augmenter, résultant en une oisiveté maximale très élevée. On peut considérer que ces missions ont échoué, puisque le SMR n'a pas assuré la surveillance régulière de tous les nœuds. De telles circonstances ne sont jamais survenues en utilisant notre approche pour anticiper l'impact des allocations sur la connectivité du RC. On peut donc dire que notre fonction d'évaluation des mises permet d'augmenter la robustesse du protocole SSI aux évènements dynamiques. Cette robustesse améliorée permet à notre approche d'être largement plus performante en moyenne.

En général, les résultats présentés dans cette section montrent qu'utiliser notre terme d'évaluation de l'évolution de la connectivité du RC pour évaluer les mises permet d'obtenir des allocations dont l'oisiveté maximale est inférieure à celles obtenues avec les autres fonction d'évaluation des mises. En particulier, dans les missions difficiles telles que celles résolues par une équipe de 10 robots, et prenant place soit sur des grilles de dimensions  $10 \times 10$ , soit sur des étoiles avec 10 branches de profondeur 5, on a montré que notre approche permet de réallouer les nœuds ayant subi des évènements dynamiques, et de visiter les nœuds au moins une fois, tandis que les autres approches n'y sont pas parvenu. Sur les missions inspirées du scénario réel de grande taille, notre approche est la seule à avoir toujours résolu tout les évènements dynamiques, tandis que les autres approches n'y sont pas toujours parvenu. En ce sens, on peut dire que l'inclusion de  $K_r$  dans la fonction d'évaluation des mises a permis d'améliorer la robustesse du protocole SSI face aux évènements dynamiques.

## 6. CONCLUSION

Dans cette étude, nous nous sommes penchés sur le problème d'allocation de tâches pour un système multi-robots dans le cadre d'une mission de surveillance. L'ATMR est un problème d'optimisation combinatoire, qui doit être résolu en temps réel et de façon distribuée. Pour ce faire, l'état de l'art est d'utiliser des méthodes basées enchère.

Nous avons proposé une formule d'évaluation des mises conçue pour inciter un protocole SSI à favoriser les allocations améliorant la connectivité du RC d'un SMR lors de la résolution de l'ATMR. Après avoir détaillé les principales étapes du protocole SSI mis en place pour résoudre l'ATMR, nous avons analysé leur complexité de calcul, et fourni des bornes sur la taille des messages qu'elles nécessitent. Nous avons ensuite présenté des simulations numériques démontrant l'intérêt de la prise en compte de l'impact des allocations sur l'évolution de la connectivité du RC dans l'évaluation des mises. En effet, ces simulations numériques montrent que notre approche permet d'améliorer la robustesse du protocole SSI face aux évènements dynamiques, et d'obtenir des allocations dont l'oisiveté maximale est meilleure que celles obtenues avec les autres approches que nous avons considérées.

Bien que nous ayons utilisé le cadre d'une mission de patrouille multi-robots pour démontrer la pertinence de notre terme de maintien de la connectivité du réseau de communication, il faut bien noter que cette pertinence s'étend au-delà des seules missions de patrouille. En effet, la formule d'évaluation des mises comprend, d'une part, notre terme de maintien de la connectivité, et d'autre part, le terme d'oisiveté, dont le rôle est d'aiguiller l'allocation des tâches vers la poursuite de l'objectif de la mission. Ce dernier terme est donc spécifique à la patrouille, et doit être remplacé pour exprimer d'autres objectifs et réaliser d'autres missions multi-robots. En revanche, le terme de maintien de la connectivité est plus général, et sa pertinence s'étend à toute application des systèmes multi-robots faisant face à des environnements dynamiques et à des communications restreintes. L'évaluation de l'apport du terme de maintien de la connectivité à d'autres cadres expérimentaux constitue le premier axe de recherche tracé par les travaux présentés dans cette article.

Pour compléter ces recherches, un second axe consiste à comparer les approches basées enchères à d'autres méthodes de résolution de l'ATMR dans le cadre de missions de patrouille multi-robots dynamiques. Un tel travail permettrait, plus largement, d'apporter un point de comparaison entre les différentes approches de résolution de l'ATMR, ce qui peut manquer en l'état actuel de la littérature. De plus, nous souhaitons explorer d'autres méthodes de résolution pour l'ATMR. En particulier, les approches multi-agent d'optimisation sous contraintes distribuées semblent prometteuses pour résoudre l'ATMR en favorisant les allocations induisant un RC bien connecté. Ces approches pourraient produire des allocations avec de meilleures performances en terme d'oisiveté, car elles permettent des recherches plus profondes de l'ensemble des solutions réalisables du problème, mais elles sont également moins rapide que les ABE. Il faudra donc trouver des critères plus stricts que ceux décrits dans cette étude pour déterminer quand procéder à des réallocations.

## BIBLIOGRAPHIE

- [1] M. A. ALSHAWI & M. B. SHALAN, « Minimal time dynamic task allocation for a swarm of robots », *International Journal of Mechanical Engineering and Robotics Research* **6** (2017), n° 6, p. 481-487.
- [2] R. C. ARKIN, T. BALCH & E. NITZ, « Communication of behavioral state in multi-agent retrieval tasks », in *IEEE International Conference on Robotics and Automation (ICRA)* (Atlanta, GA, USA), vol. 3, 1993, p. 588-594.
- [3] H. AZIZ, A. PAL, A. POURMIRI, F. RAMEZANI & B. SIMS, « Task Allocation Using a Team of Robots », *Current Robotics Reports* **3** (2022), p. 227-238.
- [4] P. CALOUD, W. CHOI, J.-C. LATOMBE, C. LE PAPE & M. YIM, « Indoor automation with many mobile robots », in *IEEE International Workshop on Intelligent Robots and Systems, Towards a New Frontier of Applications* (Ibaraki, Japan), vol. 1, 1990, p. 67-72.
- [5] Y. CHOI, Y. CHOI, S. BRICENO & D. N. MAVRIS, « Energy-constrained multi-UAV coverage path planning for an aerial imagery mission using column generation », *Journal of Intelligent & Robotic Systems* **97** (2020), n° 1, p. 125-139.
- [6] T. H. CORMEN, C. E. LEISERSON, R. L. RIVEST & C. STEIN, *Introduction to algorithms*, MIT press, 2009, 603-613 pages.
- [7] M. B. DIAS, R. ZLOT, N. KALRA & A. STENTZ, « Market-based multirobot coordination : A survey and analysis », *Proceedings of the IEEE* **94** (2006), n° 7, p. 1257-1270.
- [8] E. W. DIJKSTRA et al., « A note on two problems in connexion with graphs », *Numerische mathematik* **1** (1959), n° 1, p. 269-271.

- [9] G. FERRI, A. MUNAFO, A. TESEI & K. LEPAGE, « A market-based task allocation framework for autonomous underwater surveillance networks », in *OCEANS 2017* (Aberdeen, Scotland, UK), 2017.
- [10] M. L. FREDMAN & R. E. TARJAN, « Fibonacci heaps and their uses in improved network optimization algorithms », *Journal of the ACM (JACM)* **34** (1987), n° 3, p. 596-615.
- [11] B. P. GERKEY & M. J. MATARIĆ, « A formal analysis and taxonomy of task allocation in multi-robot systems », *The International Journal of Robotics Research (IJRR)* **23** (2004), n° 9, p. 939-954.
- [12] B. HEAP & M. PAGNUCCO, « Repeated auctions for reallocation of tasks with pickup and delivery upon robot failure », in *International Conference on Principles and Practice of Multi-Agent Systems* (Dunedin, New Zealand), 2013.
- [13] N. JOUANDEAU & Z. YAN, « Improved trade-based multi-robot coordination », in *IEEE Joint International Information Technology and Artificial Intelligence Conference* (Chongqing, China), 2011.
- [14] N. KALRA & A. MARTINOLI, « A comparative study of market-based and threshold-based task allocation », in *International Symposium on Distributed Autonomous Robotic Systems (DARS)* (Minneapolis, MN, USA), 2006.
- [15] A. KHAMIS, A. HUSSEIN & A. ELMOGY, « Multi-robot task allocation: A review of the state-of-the-art », in *Cooperative Robots and Sensor Networks*, Springer, 2015, p. 31-51.
- [16] M. T. KHAN & C. W. DE SILVA, « Autonomous and Market-Based Fault Tolerant Algorithms for Multi-Robot Cooperation », *Journal of Information Science and Engineering* **30** (2014), n° 2, p. 483-500.
- [17] M. KOES, I. NOURBAKHSH, K. SYCARA, M. KOES, K. SYCARA, I. NOURBAKHSH, M. KOES, I. NOURBAKHSH, K. SYCARA, S. D. RAMCHURN et al., « Heterogeneous multirobot coordination with spatial and temporal constraints », in *Proceedings of the 20th National Conference on Artificial Intelligence*, vol. 5, AAAI, 2005, p. 1292-1297.
- [18] L. KONG, W. HE, C. YANG, Z. LI & C. SUN, « Adaptive Fuzzy Control for Coordinated Multiple Robots With Constraint Using Impedance Learning », *IEEE Transactions on Cybernetics* **49** (2019), n° 8, p. 3052-3063.
- [19] G. A. KORSAN, B. KANNAN, B. BROWNING, A. STENTZ & M. B. DIAS, « xBots: An approach to generating and executing optimal multi-robot plans with cross-schedule dependencies », in *2012 IEEE International Conference on Robotics and Automation*, IEEE, 2012, p. 115-122.
- [20] X. LI, Z. LIU & F. TAN, « Multi-robot task allocation based on cloud ant colony algorithm », in *International Conference on Neural Information Processing*, Springer, 2017, p. 3-10.
- [21] G. LOZENGUEZ, A.-I. MOUADDIB, A. BEYNIER, L. ADOUANE & P. MARTINET, « Simultaneous auctions for "Rendez-Vous" coordination phases in multi-robot multi-task mission », in *IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technologies* (Atlanta, GA, USA), 2013, p. 67-74.
- [22] A. MACHADO, G. RAMALHO, J.-D. ZUCKER & A. DROGOUL, « Multi-agent patrolling: An empirical analysis of alternative architectures », in *International Workshop on Multi-Agent Systems and Agent-Based Simulation (MABS)* (Bologna, Italy), 2002.
- [23] M. MADHYASTHA, S. C. REDDY & S. RAO, « Online scheduling of a fleet of autonomous vehicles using agent-based procurement auctions », in *IEEE International Conference on Service Operations and Logistics, and Informatics (SOLI)* (Bari, Italy), 2017.
- [24] J. G. MARTIN, J. R. D. FREJO, R. A. GARCÍA & E. F. CAMACHO, « Multi-robot task allocation problem with multiple nonlinear criteria using branch and bound and genetic algorithms », *Intelligent Service Robotics* **14** (2021), n° 5, p. 707-727.
- [25] I. MEZEI, V. MALBASA & I. STOJIMENOVIC, « Auction aggregation protocols for wireless robot-robot coordination », in *International Conference on Ad-Hoc Networks and Wireless (ADHOC-NOW)* (Murcia, Spain), 2009.
- [26] A. R. MOSTEO & L. MONTANO, « Simulated annealing for multi-robot hierarchical task allocation with flexible constraints and objective functions », in *Workshop on Network Robot Systems: Toward intelligent robotic systems integrated with environments. Int. Conf. on Intelligent Robots and Systems*, 2006.
- [27] ———, « A survey of multi-robot task allocation », Tech. report, Instituto de Investigación en Ingeniería de Aragón (I3A), Tech. Rep., 2010.
- [28] C. NAM & D. SHELL, « An empirical study of task bundling for sequential stochastic tasks in multi-robot task allocation », Tech. report, TAMU-CSE-16-7, 2016.

- [29] M. OTTE, M. J. KUHLMAN & D. SOFGE, « Auctions for multi-robot task allocation in communication limited environments », *Autonomous Robots* **44** (2020), p. 547–584.
- [30] S. ÖZTÜRK & A. E. KUZUCUOĞLU, « Optimal bid valuation using path finding for multi-robot task allocation », *Journal of Intelligent Manufacturing* **26** (2015), p. 1049-1062.
- [31] L. E. PARKER, « Adaptive heterogeneous multi-robot teams », *Neurocomputing* **28** (1999), n° 1, p. 75-92.
- [32] C. PIPPIN, H. CHRISTENSEN & L. WEISS, « Performance based task assignment in multi-robot patrolling », in *ACM Symposium on Applied Computing* (Coimbra, Portugal), 2013.
- [33] C. POULET, V. CORRUBLE & A. EL-FALLAH SEGHRUCHNI, « Auction-based strategies for the open-system patrolling task », in *International Conference on Principles and Practice of Multi-Agent Systems (PRIMA)* (Kuching, Malaysia), 2012.
- [34] J. SCHERER & B. RINNER, « Multi-Robot Persistent Surveillance With Connectivity Constraints », *IEEE Access* **8** (2020), p. 15093-15109.
- [35] P. SCHILLINGER, M. BÜRGER & D. V. DIMAROGONAS, « Simultaneous task allocation and planning for temporal logic goals in heterogeneous multi-robot systems », *The international journal of robotics research* **37** (2018), n° 7, p. 818-838.
- [36] N. SEENU, R. M. KUPPAN CHETTY, M. M. RAMYA & J. MUKUND NILAKANTAN, « Review on state-of-the-art dynamic task allocation strategies for multiple-robot systems », *Industrial Robot: the international journal of robotics research and application* **47** (2020), n° 6, p. 929-942.
- [37] F. SEMPE & A. DROGOU, « Adaptive patrol for a group of robots », in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Las Vegas, NV, USA), 2003.
- [38] W. SHENG, Q. YANG, J. TAN & N. XI, « Distributed multi-robot coordination in area exploration », *Robotics and Autonomous Systems* **54** (2006), n° 12, p. 945–955.
- [39] R. G. SMITH, « The Contract Net Protocol : High-Level Communication and Control in a Distributed Problem Solver », *IEEE Transactions on Computers* **C-29** (1980), n° 12, p. 1104-1113.
- [40] P. B. SUJIT & R. BEARD, « Distributed sequential auctions for multiple UAV task allocation », in *American Control Conference (ACC)* (New York, NY, USA), 2007.
- [41] P. B. SUJIT & J. B. SOUSA, « Multi-UAV task allocation with communication faults », in *American Control Conference* (Montreal, Canada), 2012.
- [42] S. TRIGUI, A. KOUBAA, O. CHEIKHROUHO, H. YOUSSEF, H. BENNACEUR, M.-F. SRITI & Y. JAVED, « A distributed market-based algorithm for the multi-robot assignment problem », *Procedia Computer Science* **32** (2014), p. 1108-1114.
- [43] H. WANG & W. CHEN, « Simulated Annealing Algorithms for the Heterogeneous Robots Task Scheduling Problem in Heterogeneous Robotic Order Fulfillment Systems », in *International Conference on Intelligent Autonomous Systems*, vol. 412, Springer, 2022, p. 276-287.
- [44] X. WANG & B. SHENG, « Multi-robot task allocation algorithm based on anxiety model and modified contract network protocol », in *IEEE Information Technology, Networking, Electronic and Automation Control Conference* (Chengdu, China), 2017.
- [45] N. XI, T. J. TARN & A. K. BEJCZY, « Event-based planning and control for multi-robot coordination », in *IEEE International Conference on Robotics and Automation (ICRA)* (Atlanta, GA, USA), 1993.
- [46] C. YAN & T. ZHANG, « Multi-robot patrol : A distributed algorithm based on expected idleness », *International Journal of Advanced Robotic Systems* **3** (2016), n° 6.
- [47] W. ZHAO, Q. MENG & P. W. CHUNG, « A heuristic distributed task allocation method for multivehicle multitask problems and its application to search and rescue scenario », *IEEE transactions on cybernetics* **46** (2015), n° 4, p. 902-915.
- [48] Y. ZHENG, Y. XIAO & Y. SEO, « A tabu search algorithm for simultaneous machine/AGV scheduling problem », *International Journal of Production Research* **52** (2014), n° 19, p. 5748-5763.

**ABSTRACT.** — In the last ten years, various industrial applications of multi-robot systems have emerged. Among these are systems executing pick and delivery missions, such as the well known autonomous terminal in Rotterdam, as well as many systems tasked with surveillance missions. A key step in the execution of any multi-robot mission is the resolution of the multi-robot task allocation problem. It consists in assigning the tasks of the mission to the robots. If one needs to optimize an objective representing the performances of the system while assigning the tasks, it becomes an integer programming problem. To solve multi-robot task allocation, researchers have proposed many approaches. Among these proposals, a class of approximate methods, based on auction mechanisms, has drawn the attention of researchers for their ability to quickly reallocate tasks if this improves the execution of the mission. In this paper, we introduce a new term in the evaluation of bids for an auction based task allocation protocol. This new term enables us account for the connectivity of the communication network, which represents the communication links between the robots. The connectivity of the communication network is key to the efficiency of auction-based methods, as they need to share auction messages as broadly as possible to produce efficient allocations. We evaluated our method in a surveillance scenario. We derived theoretical bounds of the time complexity of the evaluation of the bids and of the size of the data shared during the mission. We demonstrated through simulation experiments that improved communications increase the robustness of the multi-robot system to dynamic events.

**KEYWORDS.** — Multi-Robot Systems, Task Allocation, Auction-Based Approaches, Multi-Robot Patrol.

---

*Manuscrit reçu le 18 mars 2022, accepté le 23 novembre 2022.*