

Allocation des tâches en parallèle dans un essaim de robots Etude et mise en œuvre sur la plateforme e-puck

Ing. N. BAIBOUN
ECAM – Bruxelles

Ce travail propose une méthode qui permet à un groupe de robots mobiles autonomes d'exécuter des tâches dans un ratio donné de manière auto-organisée dans un problème d'allocation des tâches en parallèle. La méthode proposée est robuste et basée sur une communication locale entre les robots. Différentes stratégies de communication sont présentées et comparées entre-elles dans un environnement simulé.

Mots-clefs : allocation des tâches, comportement collectif, robotique en essaim, intelligence en essaim, système auto-organisé, approvisionnement

This work proposes a method that allows a group of autonomous mobile robots to carry out tasks in a given ratio in a self-organized way in a parallel task allocation problem. The proposed method is robust and based on a local communication between robots. Different communication strategies are presented and compared in a simulated environment.

Keywords: task allocation, collective behaviour, swam robotics, swarm intelligence, self-organized system, foraging

1. Introduction

L'intelligence artificielle, domaine en expansion depuis quelques décennies, s'inspire de phénomènes biologiques afin de proposer des méthodes permettant de résoudre différentes classes de problèmes. Il n'existe donc pas une intelligence artificielle capable de résoudre tous les problèmes. Par exemple, la méthode MIN-MAX ne permet de résoudre que les problèmes de jeux avec adversaire [7].

L'intelligence en essaim diffère de l'intelligence artificielle classique par la philosophie à la base des méthodes utilisées. Contrairement à l'intelligence artificielle, qui vise à améliorer l'efficacité d'un individu, l'intelligence en essaim tente de garder les individus les plus simples possibles, tout en augmentant l'efficacité d'un groupe d'individus. Cette forme d'intelligence s'inspire en réalité des comportements rencontrés chez les animaux vivant en société, en particulier chez les insectes dont provient le nom de cette intelligence.

L'intelligence en essaim est utilisée pour différentes applications. Celles-ci peuvent être regroupées en 3 groupes : les animations graphiques, les algorithmes d'optimisation et la robotique collective. Cet article traitera exclusivement de la robotique collective, appelée robotique en essaim lorsque les principes de l'intelligence en essaim sont appliqués.

La robotique collective existait déjà avant l'apparition de l'intelligence en essaim. Les robots étaient tous soumis aux ordres d'un chef unique. Ce dernier centralisait les informations provenant de tous les robots afin d'allouer les ressources de façon optimale. Cependant, les temps de traitements augmentaient avec le nombre d'informations reçues, et la technologie en matière de communication limitait le périmètre d'action du groupe de robots. Les méthodes développées par l'intelligence en essaim permettent d'améliorer la redondance en supprimant l'existence d'un chef de groupe ainsi qu'à supprimer les inconvénients d'une communication globale en décentralisant l'information localement au sein du groupe.

Les robots peuvent être utilisés pour résoudre différents problèmes, par exemple, l'agrégation d'objets, la ségrégation, l'exploration, les déplacements coordonnés ou encore les manipulations coopératives. Pour

cela, les robots seront amenés à allouer les ressources afin de résoudre leurs problèmes de façon optimale. Ce dernier thème, appelé allocation des tâches, sera à la base de cet article.

L'allocation des tâches consiste à répartir les tâches de manière adéquate entre les individus composant un groupe. Dans le cadre de la robotique collective, les méthodes utilisées visent à optimiser les performances du groupe. Les travaux de A. Brutschy [3] proposent quelques méthodes de décomposition des tâches en plusieurs tâches simples, réalisables facilement par un seul individu. Ces travaux introduisent les problèmes d'allocation des tâches en parallèle, c'est-à-dire quand les différents types de tâches sont réalisés au même moment. Ce type de problèmes, facilement réalisable tel quel, peut devenir très complexe lorsqu'une contrainte supplémentaire est ajoutée. Deux types de contraintes peuvent être rencontrés : les contraintes liées à l'énergie totale du groupe ou les contraintes de ratio entre tâches. Le premier type a fait l'objet de recherches par A. Campo et M. Dorigo [4]. En ce qui concerne le second type de contraintes, peu d'études ont été menées dans ce sens jusqu'à présent. Ce travail se situe donc dans les premiers pas de la résolution de tels types de problèmes.

L'allocation des tâches en parallèle soumises à une contrainte de ratio consiste à exécuter, en même temps, différents types de tâches tout en respectant le ratio imposé. Par exemple, lors de la fabrication d'un vélo, il faut produire 2 roues pour 1 cadre. Les ratios $2/3$, $1/3$ doivent être respectés afin de continuer la production.

La suite de cet article sera structurée ainsi: la Section 2 décrira le problème traité lors de ce travail. La Section 3 proposera une méthode de résolution pour de tels problèmes. La Section 4 décrira l'environnement expérimental. La Section 5 reprendra les résultats obtenus et les discutera. Enfin, la Section 6 tirera les conclusions et proposera quelques pistes de travaux futurs.

2. Description du problème

Ce travail se focalise sur l'étude de l'allocation des tâches en parallèle au sein d'un groupe de robots autonomes devant réaliser des tâches de nature

différente en respectant un ratio exigé. Cette étude ne traitera que le cas de 2 types de tâches à exécuter en parallèle.

Dans ce cas-ci, l'environnement propose autant de tâches de chaque type. De plus, un groupe de robots non-adaptatifs, c'est-à-dire ne pouvant réaliser que les tâches d'une seule nature, est ajouté à l'environnement. La taille de ce groupe est inconnue des autres robots. La méthode doit être robuste et adaptative afin de compenser les perturbations engendrées par les robots non-adaptatifs.

3. Méthode proposée

La méthode proposée permet à un groupe de robots d'effectuer 2 types de tâches dans un ratio donné de manière auto-organisée. Chaque robot prend la décision d'effectuer une tâche de type 1 ou de type 2, juste après avoir terminé une tâche, selon les informations obtenues depuis son environnement. La méthode est entièrement distribuée et utilise une communication locale explicite entre les robots. Chaque robot envoie une information sur la dernière tâche qu'il a effectuée aux robots se trouvant dans son entourage. La méthode permet au groupe de robots d'effectuer les tâches dans le ratio imposé malgré des perturbations éventuelles.

Dans la méthode proposée, chaque robot enregistre les informations reçues dans l'ordre de réception et calcule le pourcentage de tâches effectuées de chaque type. Ce pourcentage est une approximation du ratio global basé sur la perception locale du robot. La décision est prise suivant un algorithme basé sur les méthodes de contrôle optimal.

Soit T_f la future tâche à réaliser.

$$T_f \in [T_1, T_2]$$

avec T_n : une tâche de type n

On choisit :

$$T_f : E(T_f) = \min(E(T_1), E(T_2))$$

avec E étant :

$$E(T_n) = |P(T_n) - P_n|$$

avec $E(T_n)$: erreur sur le pourcentage de tâches effectuées si la prochaine tâche choisie par le robot est de type T_n

$P(T_n)$: pourcentage de tâches de type T_n effectuées

P_n : pourcentage cible de tâches de type T_n

La méthode introduit un facteur d'oubli limitant le nombre de messages reçus et écrasant les données plus anciennes par des nouvelles. Ce facteur d'oubli permet de garder la sensibilité du robot constante une fois que la limitation du nombre de messages est activée. En effet, soit N le nombre de messages reçus. Lors de la décision du robot, le calcul du pourcentage pour chaque type de tâche ajoutant un message fictif T_f , le nombre de messages traités devient $N+1$. La sensibilité du robot à ce nouveau message vaut alors :

$$S = 1/(N + 1)$$

Cette dernière équation montre que la sensibilité du robot lors de la prise de décision décroît de manière hyperbolique avec le nombre de messages enregistrés et le robot perd sa capacité à réagir aux perturbations. Le facteur N_{\max} sera choisi expérimentalement, suivant la capacité du groupe à effectuer des tâches.

4. Environnement expérimental

L'environnement expérimental décrit dans cette section permet de valider la méthode proposée à la Section 3.

Dans ces expériences, l'environnement propose 2 types indépendants de tâches pouvant être réalisées en parallèle. Ces tâches sont localisées dans des zones prévues à cet effet à l'intérieur d'une arène fermée. L'environnement propose autant de tâches de chaque type, réparties uniformément de sorte à égaler leur probabilité d'être exécutées et ne pas créer un déséquilibre naturel entre les 2 types lors du déroulement des expériences. Les robots, après avoir pris leur décision, recherchent activement une tâche libre du type correspondant à leur décision afin de l'exécuter. Comme une communication locale est autorisée, les robots possèdent une approximation de l'activité globale du groupe.

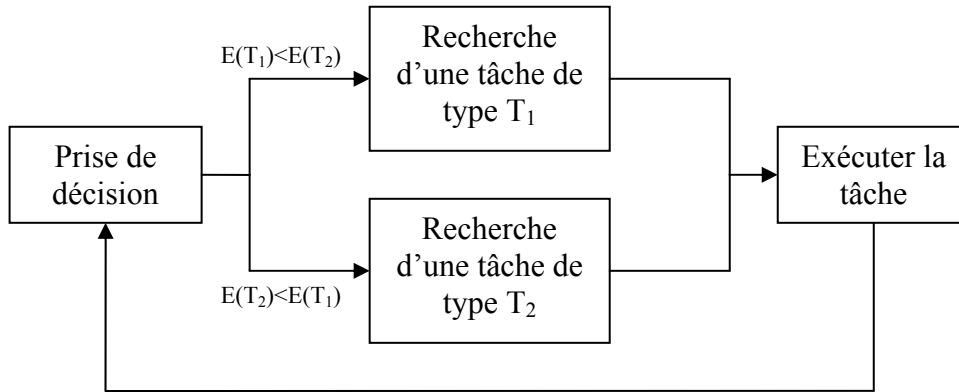


Figure 1 : Machine d'état simplifiée représentant le comportement de chaque robot. $E(T_n)$ est l'erreur définie à la Section 3 pour une tâche de type n .

4.1 Outils de simulation

Les résultats présentés dans cet article sont obtenus par l'utilisation du framework de simulation ARGoS [6]. Ce dernier est un environnement de simulation physique à temps discret permettant de simuler des expériences avec différents niveaux de détails. Pour les expériences présentées dans cet article, la simulation du mouvement plan en cinématique est suffisante. Une interface de contrôle commune fournit un accès transparent au hardware réel et simulé, permettant au même contrôleur de fonctionner sur le robot réel sans aucune modification.

Les robots utilisés lors de cette recherche sont les e-pucks¹ [5]. Ceux-ci ont été conçus afin d'être à la fois des outils de recherche et d'apprentissage pour les universités. ARGoS simule le set complet de capteurs et actuators présents sur l'e-puck. Lors de nos expériences, seuls les moteurs entraînant les roues, la caméra VGA ainsi que les 8 capteurs IR, en tant que capteurs de proximité et capteur/récepteur de messages IR, seront utilisés.

¹ <http://www.e-puck.org/>

4.2 Abstraction des tâches

Le robot e-puck étant limité par ses faibles capacités mécaniques, un dispositif a été conçu afin de simuler la réalisation des tâches par les robots e-puck. Une représentation schématique du dispositif, appelé portique [6,7], est donnée à la figure 1.

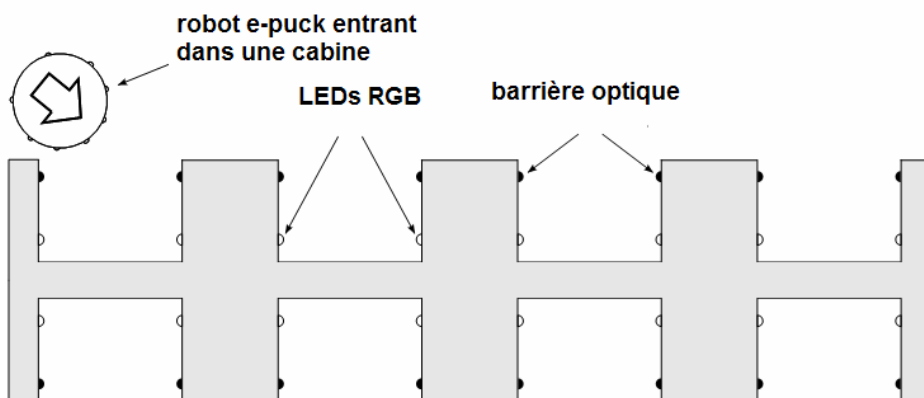


Figure 2 : Représentation schématique d'un portique comportant 8 cabines. La barrière optique, représentée par des demi-cercles noirs, permet de détecter la présence d'un robot entrant dans une cabine. Les LEDs, utilisées pour signaler la présence d'une tâche, sont représentées par des demi-cercles blancs.

Les portiques sont constitués d'un nombre variable d'emplacements, nommées cabines. Chaque cabine possède 2 LEDs RGB pouvant être perçues par les robots grâce à leur caméra couleur. Une barrière optique permet de détecter la présence d'un robot à l'intérieur d'une cabine. Tous les événements pouvant se produire sur ce dispositif sont programmables.

Durant les expériences, une couleur bleue ou verte affichée par les LEDs représente respectivement une tâche de type bleue ou verte. Ces tâches ne peuvent être effectuées qu'au sein des cabines correspondantes. Les portiques sont programmés pour afficher autant de tâches de chaque couleur et réparties uniformément dans l'environnement. Lorsqu'un robot perçoit une tâche correspondante au type choisi par son algorithme, il se dirige vers la cabine présentant cette tâche. Dès qu'il atteint la barrière optique, celle-ci

s'active et informe la cabine qu'un robot se trouve à l'intérieur. La cabine change la couleur affichée par les LEDs par une couleur rouge, informant le robot qu'il a atteint la tâche et peut procéder à la réalisation de celle-ci. Le robot reste à l'intérieur de la cabine pendant le temps requis pour réaliser la tâche. Lorsque le temps est écoulé, le robot quitte la cabine et la barrière optique informe celle-ci du départ du robot. La cabine peut alors afficher une nouvelle tâche de même couleur que la précédente.

Ces portiques permettent la réalisation d'expériences faisant abstraction de la nature réelle des tâches. Toute situation similaire peut alors être résolue par la méthode proposée dans cet article.

4.3 Environnement

L'environnement expérimental est de forme hexagonale. Cette forme est obtenue par le positionnement de 6 portiques dont seule la partie intérieure est active. L'hexagone formé est régulier et possède un diamètre de 1.6m. La forme et la dimension de l'arène permettent d'insérer de grands groupes de robots ainsi que d'éviter qu'un robot sortant d'un portique puisse directement repérer une tâche située en face de lui.

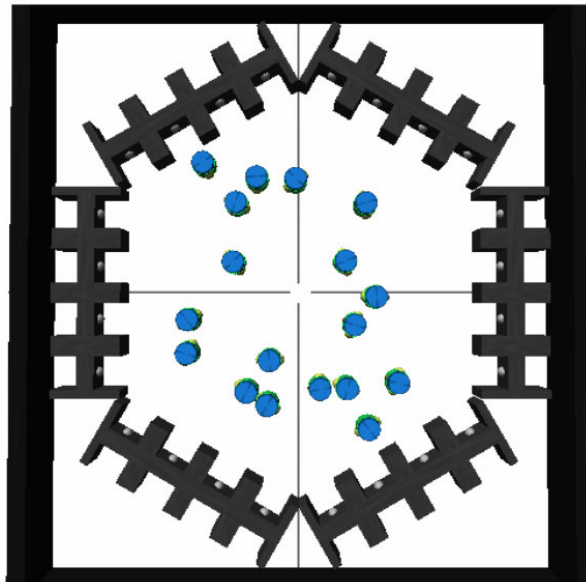


Figure 3 : Représentation de l'arène.

4.4 Paramètres expérimentaux

Les expériences lancées se focalisent sur l'adaptabilité de la méthode. Le groupe de robots est composé de 20 e-pucks, que l'on peut diviser en 2 sous-groupes. Le 1^{er} est composé de 15 robots adaptatifs pouvant choisir d'effectuer une tâche de couleur bleue ou verte suivant la méthode expliquée à la section 3. Le 2^{ème} est composé de 5 robots non-adaptatifs ne pouvant effectuer que des tâches de couleur verte. Les robots adaptatifs ne possèdent aucune information sur la taille des 2 sous-groupes de robots. Tous les robots envoient des informations.

Au sein de l'environnement décrit à la section 4.3, une comparaison est effectuée entre 3 stratégies de communication, dont les paramètres ont été choisis expérimentalement.

La 1^{ère} stratégie, appelée *stratégie simple*, consiste à ne communiquer que la couleur de la dernière tâche effectuée. Aucun filtrage n'est ajouté. Les robots peuvent donc recevoir plusieurs fois le même message, provenant du même robot. Le paramètre N_{\max} est fixé à 50 messages.

La 2^{ème} stratégie de communication, appelée *stratégie moyenne*, consiste à communiquer son propre identifiant ainsi que la couleur de la dernière tâche effectuée. Un paramètre de filtrage, appelé temps de rejet T_r , est fixé à 10s. Ce temps de rejet permet de refuser tous les messages provenant d'un même robot pendant une durée T_r suivant la réception d'un message. Après ce laps de temps, la prochaine information reçue de ce robot est acceptée. Le paramètre N_{\max} est fixé à 200 messages.

La 3^{ème} stratégie de communication, appelée *stratégie complexe*, consiste à communiquer son identifiant, la couleur de la dernière tâche effectuée ainsi que le nombre total de tâches réalisées par le destinataire. Ce dernier paramètre permet de rejeter les messages identiques. Aucun paramètre de filtrage n'est ajouté. Le paramètre N_{\max} est fixé à 200 messages.

Les robots seront soumis, pour chaque stratégie de communication, à 3 expériences différentes définissant les ratios vert/bleu suivant : 50/50, 70/30, 90/10.

Les expériences sont lancées durant une période de 10000 secondes et chaque expérience est répétée 30 fois, chaque répétition possédant des conditions initiales différentes. Les robots sont dispersés aléatoirement dans l'arène au début de chaque expérience. Comme les robots ne possèdent aucune information au début de l'expérience, la 1^{ère} tâche choisie par les robots adaptatifs est choisie aléatoirement.

5. Résultats et discussions

Comme l'environnement propose le même nombre de tâches de chaque type et qu'un groupe de robots non-adaptatifs est ajouté dans cet environnement, les robots restants doivent s'adapter à ces conditions afin d'atteindre le ratio désiré. Les informations reçues sont leur seule mesure de la performance globale réellement atteinte et seule une information correcte leur permet de prendre des décisions efficaces. Les résultats présentés dans cette section permettent de vérifier le niveau de communication nécessaire au bon fonctionnement de la méthode proposée à la section 3.

Afin de mesurer la performance du groupe de robots, les graphiques représentent le pourcentage moyen de tâches vertes effectuées dans le temps par rapport au pourcentage cible. Sur ces mêmes graphiques sont représentés les écarts-types de certains points, montrant la précision de la méthode.

La figure 4 représente les résultats obtenus pour la stratégie simple. Ces résultats montrent que les différents ratios ne sont pas atteints mais aussi que la méthode est imprécise, les écart-types étant élevés par rapport aux autres stratégies. Cette stratégie de communication rend la méthode non-adaptative.

La figure 5 représente les résultats obtenus pour la stratégie moyenne. Les différents ratios sont atteints avec une petite erreur, d'environ 5%, sur la valeur finale. De plus, la méthode est assez précise, les écart-types étant assez réduits, sauf dans le cas du ratio de 50%. Cette stratégie de communication donne donc d'assez bons résultats.

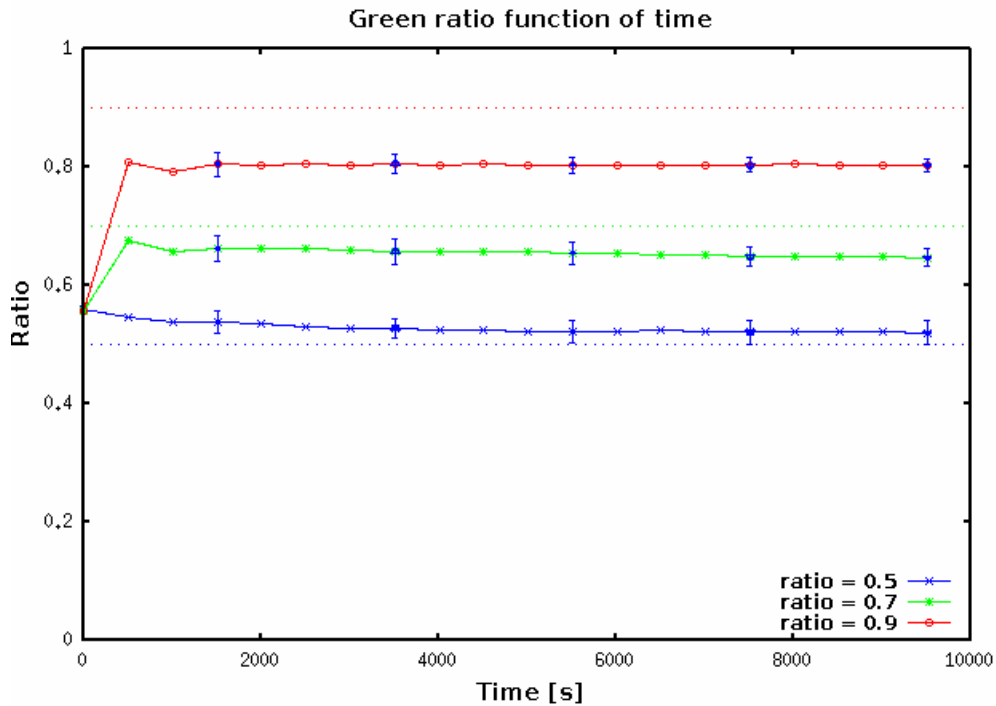


Figure 4 : Pourcentage de tâches vertes effectuées en fonction du temps dans le cas de la stratégie de communication simple. Les lignes pointillées représentent les 3 ratios cibles. Les lignes verticales représentent l'écart-type sur la mesure du ratio.

La figure 6 représente les résultats obtenus pour la stratégie complexe. Les résultats obtenus sont assez proches de ceux obtenus avec la stratégie moyenne, avec une erreur légèrement plus grande sur le ratio de 50% et des résultats un peu plus précis sur le ratio de 90%. Cette stratégie de communication rend la méthode adaptative.

Il est évident qu'une communication globale, où un ordinateur central récupère tous les messages et alloue les tâches aux différents robots via un algorithme de calcul, aurait permis d'atteindre les différents ratios pratiquement sans erreurs et avec un écart-type très réduit. Cependant, lorsque le groupe de robots devient très grand et qu'ils évoluent dans un environnement immense, la stratégie de communication globale rencontre des limites de faisabilité. Tout d'abord, la taille de l'environnement rendrait

la communication des robots les plus éloignés avec l'ordinateur impossible, mais aussi, l'ordinateur ne pourrait plus traiter toutes les données assez

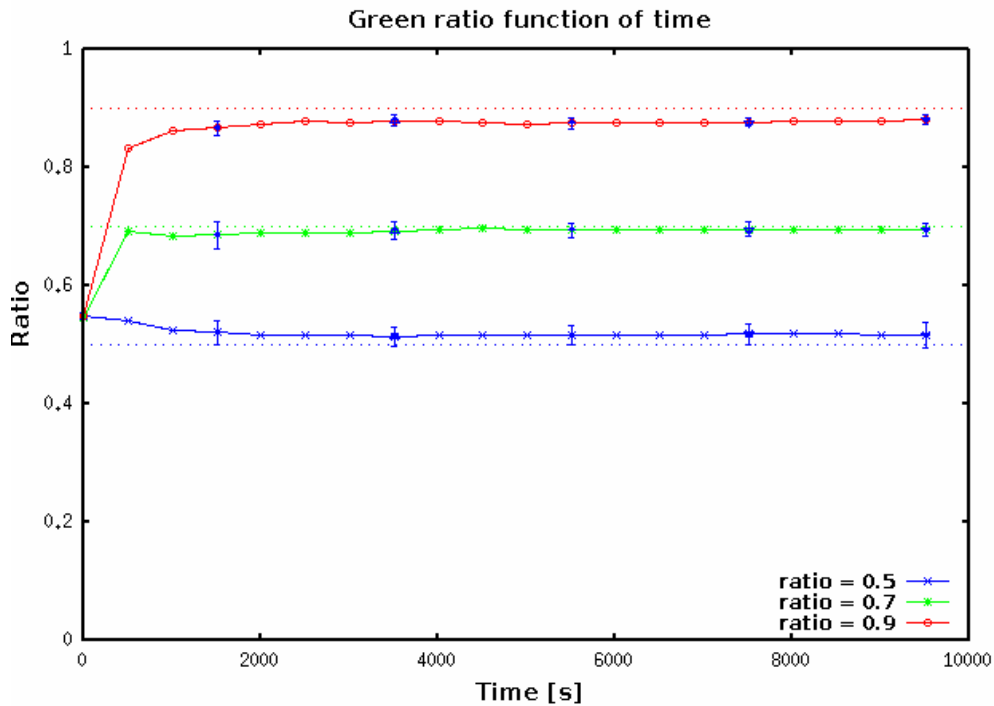


Figure 5 : Pourcentage de tâches vertes effectuées en fonction du temps dans le cas de la stratégie de communication moyenne. Les lignes pointillées représentent les 3 ratios cibles. Les lignes verticales représentent l'écart-type sur la mesure du ratio.

rapidement. De plus, une panne de l'ordinateur central arrêterait l'activité complète du groupe de robots, ce qui n'est pas le cas dans la méthode proposée dans ce travail. En effet, tous les robots étant identiques, un robot en panne est vite remplacé par ses congénères, et les informations continuent à circuler au sein du groupe. De plus, la communication étant effectuée de manière locale, le groupe de robots peut agir dans un environnement très grand sans que le manque d'information des robots les plus éloignés ne soit un frein à l'activité du groupe. La méthode proposée permet donc de pallier à ces problèmes tout en gardant une certaine adaptabilité et une certaine précision.

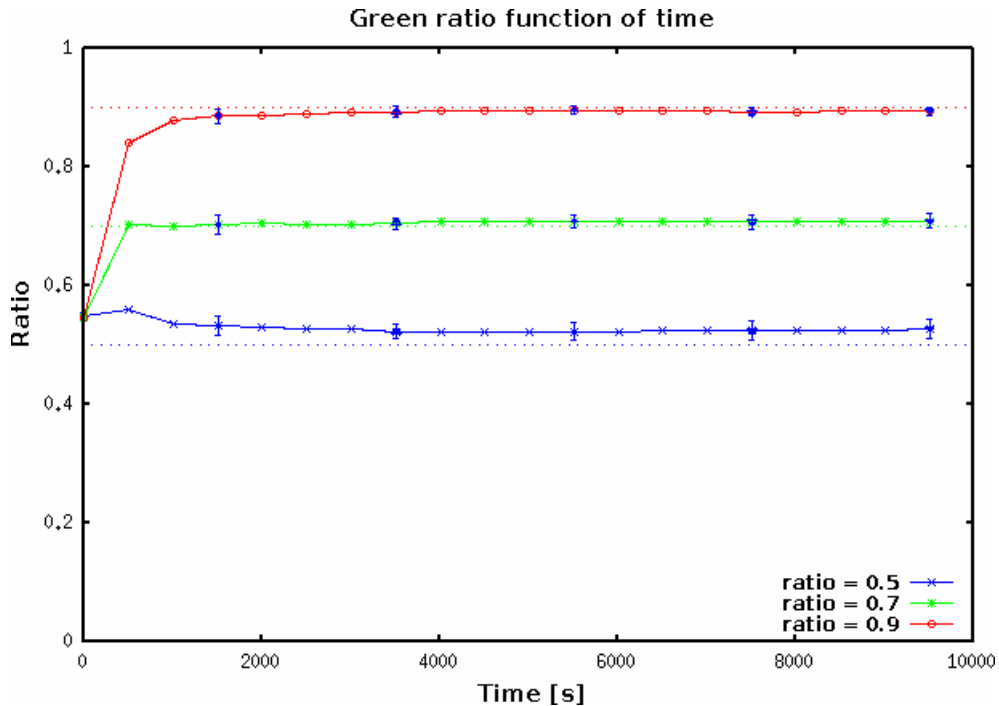


Figure 6 : Pourcentage de tâches vertes effectuées en fonction du temps dans le cas de la stratégie de communication complexe. Les lignes pointillées représentent les 3 ratios cibles. Les lignes verticales représentent l'écart-type sur la mesure du ratio.

6. Conclusions

Ce travail a proposé une méthode pour permettre à un groupe de robots autonomes de résoudre de manière auto-organisée un problème d'allocation des tâches en parallèles soumises à une contrainte de ratio. Différentes stratégies de communication ont été comparées. La méthode proposée s'est avérée adaptative avec l'utilisation de stratégies de communication moyennes et complexes, la stratégie simple étant incapable de s'adapter à différents ratios. La méthode permet aussi de résoudre des problèmes rencontrés avec des communications globales et permet une certaine redondance naturelle ainsi qu'une certaine robustesse.

Les travaux futurs consisteraient à résoudre des problèmes à plus de 2 tâches en parallèles. Un intérêt plus particulier devrait être porté sur l'algorithme de prise de décision.

7. Sources

- [1] BAIBOUN Nadir, *Rapport de stage*, ULB, 2009
- [2] BRUTSCHY Arne, PINI Giovanni, BAIBOUN Nadir, DECUGNIÈRE Antal, BIRATTARI Mauro, *The IRIDIA TAM : A device for task abstraction for the e-puck robot*, TR/IRIDIA/2010-15, ULB, 2010
- [3] BRUTSCHY Arne, *Task Allocation in Swarm Robotics*, Belgique, ULB, 2009
- [4] CAMPO Alexandre et DORIGO Marco, *Efficient Multi-Foraging in Swarm Robotics*, Advances in Artificial Life, Proceedings of ECAL 2007, Lecture Notes in Artificial Intelligence, volume LNAI 4648, 2007, pp. 696-705
- [5] MONDADA Francesco et al., *The e-puck, a Robot Designed for Education in Engineering*, Proceedings of the 9th Conference on Autonomous Robot Systems and Competitions, Portugal, 2009, pp. 59-65
- [6] PINCIROLI Carlo, *The Swarmanoid Simulator*, TR/IRIDIA/2007-025, ULB, 2007
- [7] RUSSELL Stuart et NORVIG Peter, *Artificial Intelligence – A Modern Approach*, New Jersey, Prentice-Hall, 1995, pp. 122-133

8. Remerciements

Ce travail a été partiellement supporté par le projet Swarmanoid, fondé par le Fonds de la Recherche Scientifique F.R.S.-FNRS de la communauté française de Belgique. L'auteur tient à remercier Dorigo Marco et Birattari Mauro, respectivement directeur de recherche et chercheur qualifié du F.R.S.-FNRS, pour avoir permis la réalisation de ce travail, ainsi que Brutschy Arne et Pini Giovanni, doctorants à l'IRIDIA, pour leur aide et conseils précieux.