HAL
open science

Auto-organisation d'un collectif de robots pour l'allocation dynamique de tâches

Jérôme Lacouture, Pierre Glize, Marie-Pierre Gleizes

To cite this version:

Jérôme Lacouture, Pierre Glize, Marie-Pierre Gleizes. Auto-organisation d'un collectif de robots pour l'allocation dynamique de tâches. 18ème Congrès francophone sur la Reconnaissance des Formes et Intelligence Artificielle (RFIA 2012), LIRIS : Laboratoire d'InfoRmatique en Image et Systèmes d'information, Lyon; AFIA : Association Française d'Intelligence Artificielle; AFRIF : Association Française de Reconnaissance et d'Interprétation des Formes, Jan 2012, Lyon, France. hal-00656512

HAL Id: hal-00656512

https://hal.science/hal-00656512v1

Submitted on 17 Jan 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Auto-organisation d'un collectif de robots pour l'allocation dynamique de tâches

J. Lacouture M-P. Gleizes P. Glize

Institut de Recherche en Informatique de Toulouse,
Université Paul Sabatier - Toulouse III, France

118 Route de Narbonne, F-31062 TOULOUSE CEDEX 9
{Jerome.Lacouture, Marie-Pierre.Gleizes, Pierre.Glize}@irit.fr

Résumé

L'auto-organisation et l'allocation dynamique de tâches au sein de systèmes multi-robots présentent une grande complexité lorsque ces robots sont impliqués dans des scénarios de gestion de crise. Le caractère évolutif des besoins métiers (nouvelle tâches, priorités), des contraintes de communication (perturbations) et des contraintes physiques (pannes, gestion de l'énergie) rendent évidents les besoins d'autonomie et d'adaptabilité de tels systèmes. L'auto-organisation du collectif apporte une solution par la définition de comportements locaux autonomes prenant en compte l'ensemble des contraintes citées, sans connaissance a priori de l'organisation à produire pour bien réagir à la situation de crise. Dans cet article, après avoir justifié le choix de cette approche, nous présentons le modèle de comportement des agents que nous avons implémentés au sein de la plateforme de simulation du projet RTRA STAE ROSACE (Robots). Le scénario de l'application concerne la découverte et le secours de victimes par des robots terrestres, dans des situations de feux de forêts.

Mots Clef

Auto-organisation, Adaptation, Robotique Collective, Emergence.

Abstract

Self-organization and dynamic tasks allocation in multi-robots systems takes a high complexity when adrressing crisis management scenarios. Highly changing mission needs (new tasks, priorities), evolving communication constraints (disruption) and physical constraints (breakdowns, energy management) emphasize autonomy and adaptivity requirements of such systems. Self-organization aims at designing local behaviors, taking into account quoted constraints, without any knowledge of the expected organization, in order to react to the crisis situation. In this paper, we motivate the use of this approach and we propose the behavioral model of agents we implement in the simulator of the RTRA STAE ROSACE project. The considered scenario adressed the discovery and the rescue of victims by grounded robots, in the context of forest fires.

Keywords

Self-organisation, Adaptation, Robotics, Emergence.

1 Introduction

Le projet ROSACE [1] (RObots et Systèmes Auto-adaptatifs Communicants Embarqués), financé par la fondation de coopération scientifique RTRA STAE (Sciences et Technologies pour l'Aéronautique et l'Espace), a pour but d'étudier et de développer des moyens permettant de concevoir, spécifier, implémenter et déployer un ensemble d'entités logicielles et de robots autonomes mobiles communicants et coopérants intégrant notamment des propriétés de sécurité, d'auto-préservation, des capacités d'accomplir un ensemble de missions et des capacités d'auto-adaptation dans un environnement dynamique. Le contexte considéré est celui d'entités mobiles coopérantes et opérant en situation de gestion de crise (ex : feux de forêts), dotées de moyens de communication hétérogènes et évoluants.

Dans ce contexte de situation de crise, l'environnement dans lequel évolue les robots terrestres (ATRV - All Terrain Robotic Vehicle) pour la détection et le secours de victimes est fortement dynamique. Sur le point de vue mission, la dynamicité s'exprime par de nouvelles détections de victimes plus ou moins prioritaires (la priorité étant considérée par rapport à la criticité de chaque victime). A cela, s'ajoute l'évolution des feux et des victimes qui rend la criticité elle même changeante. Sur le point de vue des communications, la mobilité, les perturbations, les pannes posent des problèmes de connectivité et de qualité de services à considérer. Enfin, d'autres contraintes changeantes comme la gestion de l'énergie des robots, d'éventuelles pannes matérielles viennent augmenter la complexité du problème à résoudre.

La possibilité d'une allocation planifiée de tâches s'avère donc difficile de par la nature dynamique et complexe du problème considéré. L'allocation dynamique de tâches dans ce type de scénario doit faire appel à un certain niveau d'auto-organisation afin d'acquérir le plus d'autonomie,

1. http ://www.irit.fr/Rosace,737

d'adaptativité et de réactivité que possible. C'est à ce titre que nous soutenons dans ce projet l'étude de divers comportements d'auto-organisation soutenus par différentes architectures :

- approche AMAS (Adaptive Multi-Agent Systems) [12], basée sur la coopération entre entités décisionnelles autonomes (cf. section 3) ;
- approche hiérarchique de Bonnet-Torres et Al. [1], basée sur la planification dynamique et la notion d'équipe de robots ;
- approche de Gascuena et al. [7], basée sur deux modèles d'agents (réactifs via des machines à états finis, et cognitifs via des règles de décision).

Dans cet article, nous nous consacrerons uniquement à la présentation de la première de ces approches (AMAS) appliquée au contexte du projet ROSACE, les deux autres stratégies étant en cours d'implémentation.

La section 2 dresse un rapide tour d'horizon de l'allocation de tâches dans le domaine des collectifs de robots afin de positionner notre vision. La section 3 présente une première stratégie d'auto-allocation appliquée aux systèmes multi-robots. L'implémentation de cette stratégie et des premiers résultats réalisés en simulation sont décrits en section 4. Enfin, nous conclurons en présentant les travaux en cours et à venir.

2 L'allocation de tâche pour la robotique collective

Dans le domaine de la robotique collective l'allocation de tâches reste un problème complexe. De nombreux paramètres sont à prendre en compte comme l'énergie des robots, les moyens de communication potentiellement perturbés, le partage des ressources (énergie, espace), la synchronisation ou encore l'évolution des priorités allouées à chaque tâche. Ces différents paramètres se positionnent comme des contraintes fortes lorsqu'on considère qu'ils évoluent au fur et à mesure du déroulement des missions. Se pose alors la question de l'efficacité d'une planification ou de la pertinence d'une solution d'allocation dynamique sans planification a priori. Nous renverrons le lecteur intéressé vers [6] et [5] pour une catégorisation autour de différents axes comme l'auto-organisation, la formalisation de la coordination et la composition d'équipes de robots. Dans la suite de cette section, nous focalisons d'avantage sur les solutions basées sur la coordination et la coopération entre robots.

Une coordination réactive, par opposition à la planification, est souhaitable pour deux raisons. D'une part, le comportement des robots est en partie imprévisible dans un environnement dynamique. D'autre part, les tâches se modifient, apparaissent et disparaissent au gré des circonstances. Le système de répartition des tâches doit s'adapter à cette incertitude plutôt que de faire des extrapolations qui sont souvent infirmées par les faits.

Différentes approches se démarquent dans la littérature ciblant l'auto-allocation pour la robotique collective. On retrouve tout d'abord une catégorie assez vaste de solutions inspirées des mécanismes d'enchères. MURDOCH [11], basé sur le principe du Contract Net Protocol, s'inscrit dans cette mouvance. Lorsqu'un robot "découvre" une nouvelle tâche par l'intermédiaire de ses capteurs ou sous la forme d'un message radio de la part d'un utilisateur, il met celle-ci aux enchères et endosse le rôle de commissaire-priseur. On retrouve, entre autres, d'autres méthodes d'allocation dédiées à la robotique collective inspirées des mécanismes de négociations (free market) [4] et d'enchères [10] [18]. Dans ces différentes approches, il s'agit de maximiser la satisfaction (le meilleur prend la tâche considérée). Ces approches règlent le problème de l'absence de connaissance a priori mais ne proposent pas explicitement de possibilités d'adaptation au vue des aléas rencontrés.

Des approches, comme celle proposée dans [2] [3], apportent, par des règles sociales de coopération, des améliorations pour la gestion de conflits entre robots, pour la gestion de la redondance dans l'accomplissement des tâches ou encore pour l'accomplissement de tâches dites "opportunistes" favorisant l'activité d'autres robots. Cependant, cette approche se fonde sur la notion de plan prédéfini pour chaque robot et s'occupe de détecter des situations problématiques à résoudre.

D'autres modèles, comme ALLIANCE [17], proposent une allocation de tâches régie par des règles de motivation. La motivation d'un robot pour une tâche augmente avec le temps jusqu'au franchissement d'un seuil déclenchant le comportement de réalisation. Si le robot met trop de temps à achever sa tâche courante, un autre robot en fait la préemption lorsque son "seuil d'impatience" a été atteint. Le calcul de coût se fait donc implicitement en cours de réalisation et est, en grande partie, fonction du temps.

D'autres travaux couramment cités, [16] [9] [15], ont depuis plus de 10 ans porté un focus particulier sur des moyens de coordination et d'auto-allocation de tâches régissant l'activité de systèmes multi-robots. Notamment, diverses stratégies ont pu être appréhendées et implémentées afin d'améliorer une certaine tolérance aux fautes et une coordination la plus efficace que possible. Ces dernières approches se veulent sans connaissance a priori sur le problème à traiter. Toutefois, afin de présenter des travaux pertinents en termes d'évaluation, nous n'écartons pas les approches de replanification (voir [1], stratégie en cours d'implémentation). En accord avec les approches présentées dans cette section, la coopération permet une auto-allocation dynamique et efficace de tâches. Des solutions basées sur la coopération, à l'instar de celles proposées dans [3] ou [16], doivent apporter l'auto-adaptation nécessaire pour prévenir et corriger les divers problèmes rencontrés, que nous qualifions (voir section 3) comme des situations de non-coopération.

Nous nous positionnons donc dans la lignée de ce type d'approche en dotant chaque robot du système de comportements auto-adaptatifs basés sur la coopération. D'une coopération la plus souvent explicite nous nous différen-

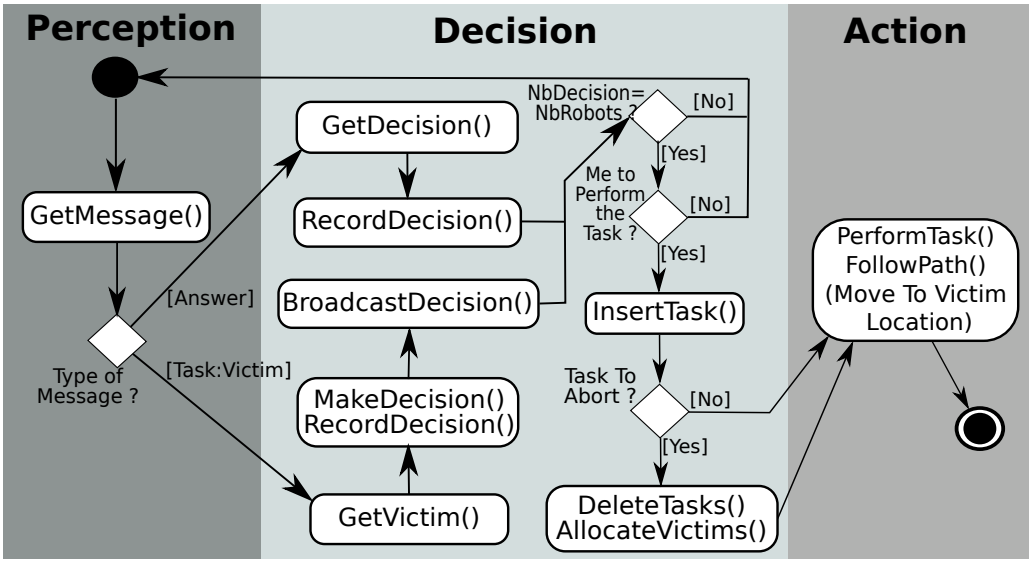

FIGURE 1 – Diagramme d'activité d'auto-allocation (AMAS) de tâches pour un robot ATRV

cions par une coopération implicite entre les différents robots du système et leur environnement qui envisage l'émergence de l'auto-allocation des tâches répondant aux besoins du système considéré, dans le cas présent à des besoins de découverte et de secours de victimes. La section suivante présente une première stratégie d'auto-allocation émergente basée sur des comportements AMAS (systèmes Multi-Agents Adaptatifs).

3 Allocation dynamique de tâches par comportement multi-agents adaptatifs

Comme nous l'avons souligné dans l'introduction, nous ne présentons dans cet article qu'une seule stratégie d'auto-allocation, la stratégie AMAS (Adaptive Multi-Agent Systems). Cette stratégie se positionne comme une stratégie de coordination réactive par opposition aux stratégies de planification (voir section 2). Dans cette section, nous présentons en premier lieu les principes des AMAS ainsi que notre vision des AMAS adaptée aux systèmes multi-robots. La section 4 permettra de mettre en oeuvre les comportements spécifiés au travers d'une architecture de robots ATRV basée sur les AMAS.

3.1 L'approche AMAS

Les comportements d'auto-allocation que nous préconisons se basent sur l'approche AMAS (Systèmes Multi-Agents Adaptatifs)[8]. Cette approche basée sur la coopération entre agents se fonde sur l'idée que chaque agent n'a pas besoin de connaissances explicites sur le problème à traiter et possède ses propres organes de perception, décision et action. La coopération sert de support à l'auto-organisation. Le concepteur ne s'intéresse qu'à l'agent et lui donne les moyens (basés sur des compétences, des croyances, des attitudes sociales, un langage d'intéraction et des aptitudes) de décider de manière autonome de changer ses "liens" avec les autres agents du système pour tendre vers une organisation coopérative. Ainsi en fonction des interactions qu'a l'agent avec l'environnement et les autres agents, l'organisation entre ces agents émerge et constitue une réponse pour faire face aux imprévus. Chaque agent essaye d'éviter des situations dites "non-coopératives". Par exemple, dans le cas d'agents "fourmis" explorant des zones à la recherche de nourriture, on essaiera d'éviter des situations non-coopératives de concurrence (où toutes les fourmis s'oriente et s'intéresse au même ensemble de ressource) en préférant l'exploration de zones pas encore exploitées par d'autres agents fourmis [13]. Au final, une solution globale du problème considéré émerge des comportements locaux. Nous renvoyons le lecteur intéressé vers [8] pour un descriptif complet de la théorie AMAS.

Dans le cadre de systèmes multi-robots, chaque robot est vu comme un agent. Les robots (qui n'ont en aucune manière besoin d'être homogènes) fondent leurs raisonnements locaux sur les mêmes critères d'évaluation. Ces critères dépendent du domaine d'application (ici le secours à des victimes humaines) et sont définis par des experts. Le traitement de situations de non-coopération correspond à la volonté d'éviter le traitement de victimes non ou moins prioritaires avant celles dites prioritaires. La prise en compte des problèmes de consommation énergétique, de distances entre robots et victimes sont aussi des problèmes à considérer comme des situations de non-coopération. Enfin l'auto-allocation globale de l'activité émerge des allocations locales autonomes sans explicitation a priori. Dans la section 3.2, nous décrivons de manière plus dynamique et plus concrète les comportements AMAS spécifiés dans le cadre du projet ROSACE.

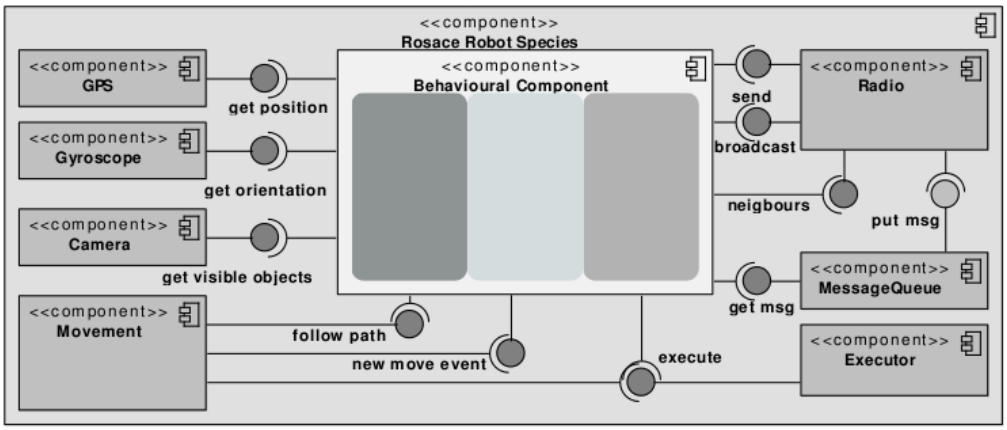

FIGURE 2 – Architecture de l'espèce robot (ATRV) produite à l'aide de SpEArAF

3.2 Comportements AMAS

Traduit sous forme d'algorithme, un comportement AMAS de robot dédié au secours de victimes peut être vu sous la forme du diagramme d'activité présenté figure 1. On retrouve dans ces comportements la volonté de séparer les processus de perception, de décision et d'action :

- Au niveau de la perception : un robot perçoit via ses capteurs (GPS, caméra, ...) et via des messages. Un robot est informé de la présence d'une nouvelle victime (position, criticité,...) par la réception d'un message diffusé par un centre de contrôle. Il est informé aussi, par diffusion de messages des autres robots, des coûts (*answers*) de prise en charge des victimes par les autres robots ;
- Au niveau de la décision : il s'agit pour le robot de définir un coût associé à l'allocation d'une tâche lorsqu'il reçoit une nouvelle victime et de stocker puis définir s'il peut être le meilleur pour réaliser une tâche lorsqu'il reçoit les coûts envoyés par les autres robots. L'évaluation d'un coût par un robot se base en premier lieu sur la notion d'abandon de tâches. Un robot dans l'obligation d'abandonner une ou plusieurs tâches pour l'accepation d'une nouvelle détermine un coût associé au niveau de criticité des tâches à abandonner. Plus le ou les tâches à abandonner sont critiques, plus le coût sera élevé. Un robot doit abandonner une tâche si, par exemple, l'acceptation d'une tâche prioritaire (secours d'une victime de haute priorité) engendre l'abandon d'une tâche moins prioritaire (secours d'une victime de priorité plus basse) du fait que l'énergie que possède le robot est insuffisante pour l'accomplissement de l'ensemble des tâches. D'autres paramètres comme le temps nécessaire pour secourir une victime peut influer sur l'abandon d'une tâche. Le calcul énergétique, dans sa version la plus simpliste basé sur la distance entre robot et victime, contribue aussi à la décision (et peut être bloquant si l'énergie dont dispose le robot n'est pas suffisante).
  La décision de s'auto-allouer une tâche se fait après réception des différents coûts émis par le collectif de robots. L'agent robot sait alors si il est le meilleur, en termes de coût, pour accomplir une tâche. Si il est le meilleur, alors la nouvelle tâche est ajoutée à la liste des tâches à effectuer (trier par ordre de priorité). Et, si il y a lieu des tâches sont abandonnées en vue d'être diffusées, via le même processus à l'ensemble des robots.
- Au niveau de l'action, chaque robot suit un chemin (*followpath()*) correspondant aux positions des différentes tâches (victimes) qu'il a pris en charge. Il effectue des actions (simulées) de secours de victime. Enfin, il envoie les messages (diffusion de coûts,. . .) aux autres robots.

Des situations non-coopératives de l'ordre de la communication peuvent venir perturber le fonctionnement nominal des comportements d'auto-allocation ainsi spécifiés. Les perturbations de communication (pannes, dégradation de la communication, perte de la connectivité) sont considérées comme fréquentes dans le type de scénario envisagé (situation de crise). On ne peut considérer alors comme acquis le fait de posséder une connectivité totale entre l'ensemble des robots. Les distances entre chaque robot et leurs mouvements ne permettent pas, en effet, de valider la possibilité d'acheminer chaque message par une simple diffusion. Si l'on peut supposer d'une communication *permanente* entre robots et centre de contrôle, il n'en est pas de même entre robots. La diffusion des coûts peut se révéler incertain. Une situation de non-coopération peut alors se manifester où des robots n'ayant pu recevoir les coûts de l'ensemble du collectif se considèrent comme les "meilleurs" pour réaliser une tâche donnée, entraînant ainsi une baisse de performance globale du système. La solution retenue réside dans la mise à jour des meilleurs coûts reçus lors de nouvelles connectivités. En d'autres termes, lorsqu'un robot entre en connectivité avec un autre, qui n'était pas précédemment dans sa portée de communication, les robots s'échangent la liste courante des meilleurs coûts pour chaque tâche. Ainsi, ce risque de situations non-coopérative, lié à des doublons dans l'allocation de tâches, est diminué.

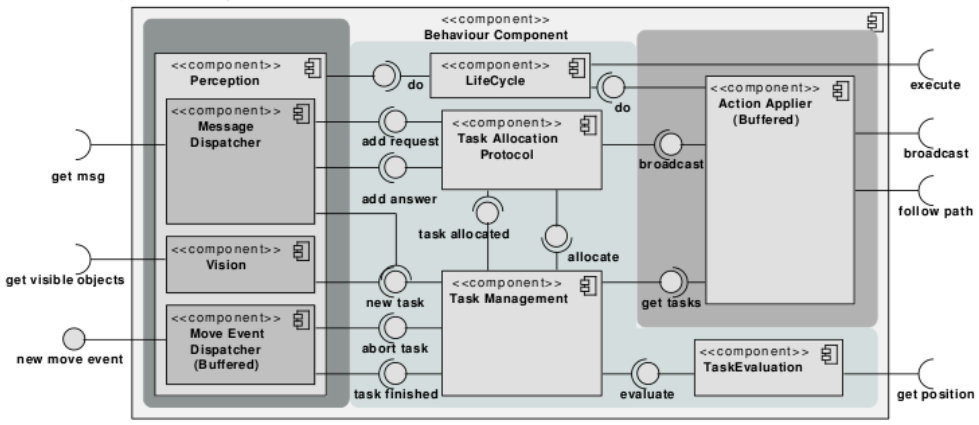

FIGURE 3 – Architecture de la sous-espèce robot (ATRV) doté de comportement AMAS produite à l'aide de SpEArAF

4 Implémentation et Résultats

Dans le cadre du projet Rosace, et comme expliqué dans cet article, nous avons décidé d'implémenter en premier lieu les comportements AMAS décrits en section 3. Le simulateur 3D Morse [2], nous permet de mettre en place et de tester nos premières implémentation avant de passer à des expérimentations sur robots réels.

4.1 Processus de développement SpEArAF

L'implémentation de robots dotés de comportements d'auto-allocation de tâches s'appuie sur le processus de développement SpEArAF [14]. Ce processus de développement, basé sur des propriétés de réutilisation et de flexibilité s'est imposé par le souci de développer différents comportements et de les comparer. Dans SpEArAF, le concept d'espèce permet la spécification de robots partageant une structure commune (ensemble de composants : radio, GPS, roues, ...) et le concept de sous-espèce autorise de dériver une espèce partageant une spécialisation ou un ensemble de modifications communes. Ce processus de développement facilite grandement la tâche du développeur. A partir d'une espèce (robot), il devient aisé de développer différentes sous-espèces correspondantes à chaque comportements que nous souhaitons expérimenter. L'architecture produite (figure 2), donne l'ensemble des composants formant un robot ATRV. Le composant *Behavior* est en charge de l'orchestration interne (entre les fonctions des différents composants) afin de garantir un comportement global cohérent.

La figure 3 correspond à la sous-espèce "robot AMAS" produite à l'aide de SpEArAF. L'architecture à composants produite respecte le même découpage que celui présenté dans la figure 1, avec perception, décision et action. En pratique, les architectures produites sont définies à l'aide de l'outil MAY (Make Agents Yourself [3], basé sur des méthodes d'ingénierie dirigée par les modèles et sur des éditeurs. Les frameworks, quant à eux, sont implémentés en JAVA.

Nous renvoyons le lecteur intéressé à [14], pour une description plus complète du processus de développement SpEArAF. Nous focalisons, par la suite, sur les résultats des expérimentation produites.

4.2 Expérimentations

L'implémentation réalisée, via l'architecture produite, met en oeuvre les comportements AMAS au sein du simulateur Morse. La figure 4 présente un aperçu de l'environnement 3D et de la manière dont sont modélisés les robots. Chaque robot est vu comme un ensemble de composants, notamment de capteurs et d'actionneurs, avec lesquels on peut interagir soit pour collecter des données, soit pour agir (mouvements du robot, secours de victimes,...).

FIGURE 4 – Robot terrestre ATRV développé dans la plateforme de simulation Morse

Dans les expérimentations présentées ici, nous prenons uniquement en compte les tâches de secours de victime, en occultant la phase d'exploration/détection de ces victimes. Afin de simplifier les expérimentations présentées, nous considérons que la flotte de robots est informée, par le centre de contrôle, par diffusion de messages de la position et de la criticité de chaque victime. Nous nous sommes volontairement placés dans une "situation trop simpliste"

2. http ://morse.openrobots.org
3. http ://www.irit.fr/MAY

afin de focaliser le sujet sur la stratégie d'allocation dynamique. Les contraintes sont vues comme des situations non-coopératives à éviter ou comme des composantes de la fonction d'évaluation. Toutefois, dans nos simulations, nous pouvons prendre en compte dès à présent les problèmes de connectivité. Le choix d'une connectivité constante entre centre de contrôle et robots s'explique par des moyens/mode de communication étendus (ex : GSM Vs WiFi entre robots). De plus, cette hypothèse a pour but de ne pas évoquer la recherche systématique de victimes par drones ou opportuniste par robots qui ne sont actuellement pas supportés par la plateforme. Les données échangées entre les robots correspondent aux coûts associés aux différentes tâches. Ainsi, le volume des communications est dérisoire car il correspond à une centaine d'octets par robot. Le processus est réitéré si cela implique de se décharger d'une tâche précédemment allouée. Comme les tâches délaissées sont de moins en moins critiques, ce processus se termine nécessairement et est au pire (très exceptionnellement) réitéré autant de fois que de victimes. Une évaluation par robot est de quelques millisecondes, ce qui fait qu'une réallocation totale serait inférieure à la seconde, ce qui est véritablement négligeable en regard du temps réel de l'application.

Une simulation commence par la découverte d'une victime. Chaque autre victime est découverte au fur et à mesure de la simulation et les robots ATRV sont informés par diffusion de leurs positions et de leurs criticités. Les robots s'auto-organisent selon la stratégie AMAS (figure 1) par coopération pour l'allocation des tâches de secours de victimes en prenant en compte leur énergie, la distance, le coût d'allocation, ...

Dans cet article, nous présentons les résultats de deux simulations réalisées dans le simulateur Morse. Une vidéo est disponible montrant un extrait d'une mission [4]. Les deux simulations mettent en jeu un collectif de 5 robots terrestres ATRV et un ensemble de 18 victimes aléatoirement disposées sur une scène comportant des obstacles (bâtiments, arbres,...). L'ordre de découverte, la criticité et l'emplacement des victimes sont des paramètres que nous faisons évoluer d'une simulation à une autre afin d'évaluer les qualités de la stratégie implémentée. En effet, la variabilité de ces paramètres peut influencer les résultats. Nous noterons que les victimes ont potentiellement besoin de soins différenciés (10 types différents dans les simulations) que peuvent apporter uniquement certains robots qui ont les compétences appropriées. Seuls ces robots-là répondront qu'ils sont pertinents pour le secours. Cette possibilité n'est pas développée dans l'article car cela n'apporte rien sur le fond de la stratégie d'allocation

Dans la première expérimentation (figure 7), des victimes de haute criticité sont découvertes et diffusées à la flotte de robots dès le début de la simulation, puis de manière aléatoire. Dans la seconde simulation, des victimes de haute criticité ne sont pas détectées au début de la simulation

4. http ://www.irit.fr/Rosace

v1	v2	v3	v4	v5	v6	v7	v8	v9	v10
9	8	9	7	2	5	1	1	2	10

v11	v12	v13	v14	v15	v16	v17	v18
8	1	1	10	5	9	7	4

FIGURE 5 – Simulation 1 : Ordre d'arrivée et priorité des victimes

v1	v2	v3	v4	v5	v6	v7	v8	v9	v10
2	2	1	10	8	10	4	5	2	8

v11	v12	v13	v14	v15	v16	v17	v18
4	2	10	7	8	10	8	9

FIGURE 6 – Simulation 2 : Ordre d'arrivée et priorité des victimes

mais sont plutôt dans les dernières détectées.

Les tableaux 5 et 6 donnent les priorités des différentes victimes dans l'ordre de détection et secours.

4.3 Résultats

Les résultats de ces simulations sont présentés selon une courbe de criticité correspondant à la somme des criticités présente dans le système à un instant T (somme des criticités associées à chaque victime détectée). Le choix du paramètre *criticité* s'est imposé de par le souci d'évaluer l'efficacité de l'approche décrite pour le secours des victimes prioritaires, ce qui correspond à un but affiché pour la résolution de situation de crise dans ROSACE.

Les criticités individuelles de chaque victime sont valuées de 1 à 10 (10 correspondant à la valeur la plus critique). Les simulations commencent par la découverte d'une première victime et par l'envoi des informations de criticité et de localisation au collectif de robots.

Afin de proposer une évaluation comparative, et dans l'attente des implémentations des autres stratégies (section 1), nous comparons nos résultats à l'implémentation d'une stratégie d'auto-allocation simpliste, basée sur la proximité des robots avec les victimes mais sans prendre en compte le niveau de criticité (courbes non-AMAS). Il est bien entendu qu'il s'agit d'une comparaison par rapport à une stratégie factice, dans l'attente d'une évaluation comparative avec d'autres stratégies en cours d'implémentation. Cependant, le but affiché ici est seulement de montrer un gain réel de par l'emploi de ce type de stratégie.

En termes de résultats, les courbes augmentent avec la détection de nouvelles victimes (le pic maximal est atteint après 3min20s de simulation : dernière victime détectée). L'augmentation de la criticité peut être atténuée par le début des secours des victimes (victimes secourues alors qu'elles n'ont pas encore été toutes détectées) ; c'est le cas, par exemple, des victimes v1 ou v3 dans la première simulation.

Pour compléter la présentation des résultats issus de la simulation 1 (figure 7), nous donnons ci-dessous la répartition des victimes pour les comportements AMAS dans l'ordre

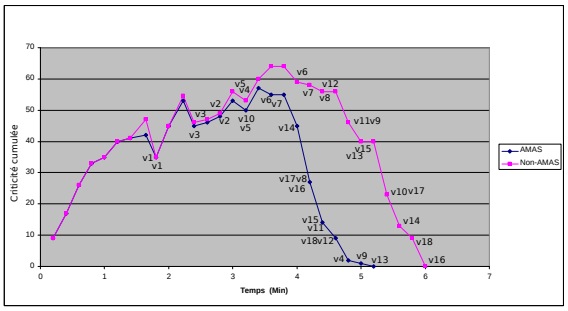

FIGURE 7 – Simulation 1 : Evolution de la criticité lors d'une mission.

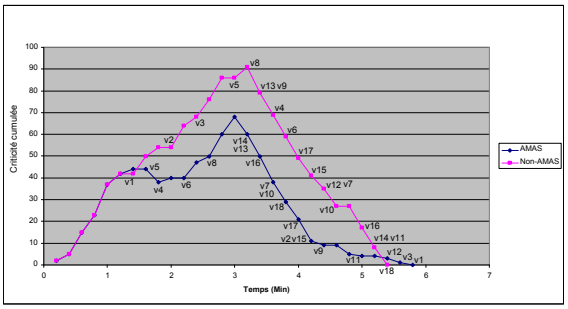

FIGURE 8 – Simulation 2 : Evolution de la criticité lors d'une mission

chronologique de secours :

- Robot1 : v5, v17, v12
- Robot2 : v1, v2, v10, v16, v4
- Robot3 : v9, v14, v11
- Robot4 : v6, v15, v9
- Robot5 : v7, v8, v18, v13

Cette répartition tient en compte les diverses ré-allocations, suite à divers abandons de tâches. La victime v9 a par exemple était allouée dans un premier temps au robot2. Afin d'accepter la tâche correspondant à la victime v10 (de plus haute priorité), la victime v9 a été abandonnée, diffusée et réallouée au robot4.

De la même façon, la répartition des victimes pour les comportements non-AMAS dans l'ordre chronologique de secours est celle décrite ci-après :

- Robot1 : v5, v12, v13
- Robot2 : v1, v2, v4, v9, v17, v18
- Robot3 : v3, v11, v14, v16
- Robot4 : v6, v15
- Robot5 : v7, v8, v10

Nous notons qu'une des différences réside dans le fait que les robots, comme le robot 2, peuvent se retrouver avec un nombre de victimes plus important dû à la proximité des victimes entre elles, ce qui a pour effet de retarder le secours des victimes, et par conséquent la baisse de la criticité et la fin de la simulation.

Autre constat intéressant, les victimes les plus prioritaires sont secourues plus rapidement avec la stratégie AMAS. Cela s'explique par le fait que les tâches de criticité haute sont traitées prioritairement, d'où l'efficacité de la stratégie. Pour donner une idée, après 4min20s de simulation, toutes les victimes de haute criticité (>=8) ont été traitées avec la stratégie AMAS (contre 5min40s sans cette stratégie).

La deuxième simulation (figure 8), permet de constater les différences possibles dûes à l'ordre de la détection de victimes plus ou moins prioritaires. Dans la première simulation, la détection de victimes de criticité haute dès le début de la simulation atténue la visibilité de l'efficacité de la stratégie AMAS car les premières victimes sont secourues dans le même ordre, ce qui a pour effet une diminution commune de la criticité globale. La stratégie AMAS prend son intérêt dans la suite de la simulation par une diminution plus rapide de la criticité.

La seconde simulation, avec la détection des victimes de haute criticité plus réparties, voire retardée, met en avant une diminution globale de la criticité cumulée. Le pic de criticité est nettement inférieur par rapport à une stratégie d'allocation basée sur la distance. En revanche, l'éloignement de certaines victimes de très basse criticité retarde la fin de la simulation (secours de l'ensemble des victimes). Cela n'est cependant pas un critère d'efficacité selon nos premières considérations. Si l'on se réfère aux secours de victimes de haute criticité (>=8), avec la stratégie AMAS, toutes les victimes de haute criticité sont secourues au bout de 4min10s contre 5min pour la stratégie de référence.

Ces premiers résultats nous confortent dans l'idée que l'approche d'auto-allocation AMAS, basée sur la coopération et la résolution de situations non-coopératives (gestion de priorités, communications, gestion de l'énergie), présente une réponse pertinente pour la résolution de situation de crise et l'allocation dynamique de tâches dans les systèmes multi-robots.

D'autres scénarios, faisant varier le nombre de victimes, leur criticité ont été expérimentés. Plus le nombre de victimes à secourir augmente, plus les résultats sont probants. Ces résultats seront à comparer avec ceux issus des implémentations des autres stratégies évoquées dans l'introduction.

5 Conclusion

Dans cet article, nous avons présenté une approche d'auto-allocation de tâches pour systèmes multi-robots. La solution spécifiée et implémentée se base sur la coopération entre robots autonomes mobiles et incluent des mécanismes d'allocation de tâches, d'évaluation du coût de l'allocation, d'abandon de tâches et d'évitement de situations non-coopératives. Les premiers résultats issus d'expérimentations effectuées sous simulateur 3D sont encourageants en termes d'efficacité du traitement des tâches prioritaires mais restent encore à être comparés à d'autres stratégies. Nos travaux en cours et à venir vont permettre d'améliorer encore cette stratégie d'auto-organisation, surtout au niveau de la prise en compte des perturbations

de la communication. L'implémentation des autres stratégies présentées en introduction permettront une évaluation comparative pertinente afin de mettre en avant les éventuels points faibles de chacune, notamment quels scénarios sont les plus problématiques en termes d'efficacité pour chacune des stratégies. De nouvelles métriques d'évaluation seront introduites pour évaluer aussi bien les performances fonctionnelles qu'opérationnelles (énergie restante pour chaque robot, répartition du nombre de victimes sauvées, ...).

6 Remerciements

Ce travail a été soutenu par le Réseau Thématique de Recherche Avancée Sciences et Technologies pour l'Aéronautique et l'Espace (RTRA STAE) dans le cadre du projet ROSACE (Robots et Systèmes Auto-adaptatifs Communicants Embarqués).

Références

[1] Olivier Bonnet-Torres and Catherine Tessier. A Formal Petri Net Based Model for Team Monitoring. *Handbook of Research on Multi-Agent Systems : Semantics and Dynamics of Organizational Models*, 23 :568–590, 2009.

[2] Silvia BOTELHO and Rachid ALAMI. M+ : a scheme for multi-robot cooperation through negotiated task allocation and achievement. *IEEE International Conference on Robotics and Automation*, pages 1234–1239, 1999.

[3] Sylvie BOTELHO and Rachid ALAMI. A multi-robot cooperative task achievement system. In *in Proceedings of the IEEE International Conference on Robotics and Automation (ICRA*, 2000.

[4] M Bernardine DIAS and Anthony STENTZ. A free market architecture for distributed control of a multirobot system. In *6th International Conference on Intelligent Autonomous Systems (IAS-6)*, pages 115–122, July 2000.

[5] G. Dudek, M. Jenkin, and E. Milios. A taxonomy of multirobot systems. *Robot Teams : From Diversity to Polymorphism*, pages 3–22, 2002.

[6] Alessandro Farinelli, Ro Farinelli, Luca Iocchi, and Daniele Nardi. Multi-robot systems : A classification focused on coordination. *IEEE Trans. Syst., Man, Cybern. B*, 34 :2015–2028, 2004.

[7] José Manuel Gascueña, Antonio Fernández-Caballero, and Francisco J. Garijo. Using ICARO-T Framework for Reactive Agent-Based Mobile Robots. In *PAAMS*, pages 91–101, 2010.

[8] Jean-Pierre Georgé, Marie Pierre Gleizes, and Pierre Glize. Conception de systèmes adaptatifs à fonctionnalité émergente : la théorie amas. *Revue d'Intelligence Artificielle*, 17(4) :591–626, 2003.

[9] Brian P. Gerkey and Maja J Mataric. Pusher-watcher : An approach to fault-tolerant tightly-coupled robot coordination. In *IEEE International Conference on Robotics and Automation (ICRA 2002)*, 2001.

[10] Brian P. GERKEY and Maja J MATARIĆ. Sold ! : Auction methods for multi-robot coordination. *IEEE Transactions on Robotics and Automation*, 18(5) :758–768, oct 2002.

[11] Brian P. Gerkey and Maja J. Mataric. A formal analysis and taxonomy of task allocation in multi-robot systems. *I. J. Robotic Res.*, 23(9) :939–954, 2004.

[12] Marie-Pierre Gleizes, Valérie Camps, Jean-Pierre Georgé, and Davy Capera. Engineering systems which generate emergent functionalities. In Danny Weyns, Sven A. Brueckner, and Yves Demazeau, editors, *EEMMAS 2007*, volume 5049 of *LNCS (LNAI)*, pages 58–75. Springer, 2008.

[13] Pierre Glize. Do artificial systems need random for learning strategies ? *International Journal of Computing Anticipatory Systems*, 1 :189–199, 1998.

[14] Jérome Lacouture, Victor Noel, Jean-Paul Gleizes Arcangeli, and Marie-Pierre. Engineering agent frameworks :an application to multi-robot systems. In *9th Internationam Conference on Practical Applications of Agents and Multiagent Systems, Salamanca Spain*, pages 79–86, 2011.

[15] Maja J Mataric. Issues and approaches in design of collective autonomous agents. *Robotics and Autonomous Systems*, 16 :321–331, 1995.

[16] Maja J. Mataric, Gaurav S. Sukhatme, and Esben H. Ostergaard. Multi-robot task allocation in uncertain environments, 2003.

[17] Lynne E. Parker. Alliance : An architecture for fault tolerant multirobot cooperation. *IEEE Transactions on Robotics and Automaton*, 1998.

[18] R. Zlot. *An Auction-Based Approach to Complex Task Allocation for Multirobot Teams*. PhD thesis, Carnegie Mellon University, 2005.