

---

# Évaluation d'un système multirobot

## cas d'une constellation de satellites

Grégory Bonnet\* — Catherine Tessier\*\*

\*Institut Charles Delaunay - Université de Technologie de Troyes  
Équipe Environnement des Réseaux Autonomes  
12, rue Marie Curie, BP 2060 - 10010 Troyes Cedex  
gregory.bonnet@utt.fr

\*\*Onera - DCSD  
Unité de Recherche Conduite et Décision  
2, avenue Edouard Belin, BP 74025 - 31055 Toulouse Cedex 4  
catherine.tessier@onera.fr

---

*RÉSUMÉ. Les systèmes multisatellites sont des systèmes multiagents particuliers dans la mesure où l'ajout d'un agent satellite est extrêmement coûteux. C'est pour cette raison que l'évaluation d'un système multisatellite par le biais de simulations est fondamentale. Il est cependant nécessaire de définir des mesures pertinentes permettant de dégager un compromis entre performances et taille du système. Dans cet article, nous présentons un retour sur l'évaluation d'algorithmes de coordination décentralisée pour une constellation de satellites autonomes d'observation. Nous définissons pour cela une notion d'expérience et des métriques relatives aux notions de performance, stabilité et capacité d'extension avant d'analyser les résultats des simulations.*

*ABSTRACT. Multisatellite systems are particular multiagent systems in so far as adding a satellite agent into the system is very costly. Therefore evaluating a multisatellite system through simulations is a major issue. However it is of great importance to set relevant measures so as to be able to make a trade-off between the efficiency and the size of the system. A practical case is presented in this paper: the evaluation of decentralized coordination algorithms for a constellation of autonomous observation satellites. An experiment notion and metrics concerning performance, stability and scalability are defined and the corresponding results are discussed.*

*MOTS-CLÉS : évaluation, capacité d'extension, métriques, systèmes multiagents.*

*KEYWORDS: evaluation, scalability, metrics, multiagent systems.*

---

## 1. Introduction

Résoudre un problème par un système multiagent permet, dans un premier temps, de n'affecter à chaque agent qu'une partie du problème et, dans un second temps, de transformer des contraintes globales en contraintes locales qui sont satisfaites par coordination entre les agents. Augmenter le nombre d'agents dans le système permet, la plupart du temps, de résoudre le problème plus facilement.

Cependant, lorsqu'il s'agit d'agents physiques tels que des robots ou des satellites, le coût de la mise en œuvre des agents est un paramètre prépondérant : ajouter un satellite à une constellation est extrêmement coûteux ; par ailleurs, comme les satellites sont limités en capacités de communication et de calcul, implanter des raisonnements embarqués complexes est très coûteux aussi. C'est pourquoi il est important de savoir s'il est intéressant d'accroître le nombre de satellites au sein d'une constellation et/ou d'accroître leurs capacités de raisonnement.

Dans le domaine spatial, le contrôle d'une constellation de satellites à l'aide d'un système multiagent est étudié depuis plusieurs années. (Schetter *et al.*, 2003) proposent une approche centralisée où un satellite unique génère un plan par construction de but pour l'ensemble de la constellation et communique les plans partiels aux autres satellites. En cas de défaillance du satellite-maître, un autre satellite peut prendre le relais. Pour réduire l'espace de recherche de cette approche, (Brambilla *et al.*, 2005) proposent une planification par construction de réseaux hiérarchiques de tâches. Cependant, ces approches n'échappent pas à l'écueil de la centralisation qui induit une fragilité en cas de défaillance et réduit la réactivité de la constellation. Une approche décentralisée a été proposée par (Damiani *et al.*, 2005) mais elle ne tient pas compte de communications entre les satellites. Nous avons alors proposé dans (Bonnet *et al.*, 2007a) une approche décentralisée fondée sur des satellites communicants.

Nous proposons d'évaluer les performances d'une telle constellation simulée en fonction de paramètres tels que le nombre de satellites et leurs capacités de raisonnement. Ceci nous conduit à définir des mesures afin de mettre en évidence le compromis entre performance et nombre d'agents satellites. Pour cela, nous introduirons des mesures spécifiques au domaine multiagent telles que la capacité d'extension<sup>1</sup> et la capacité de coopération.

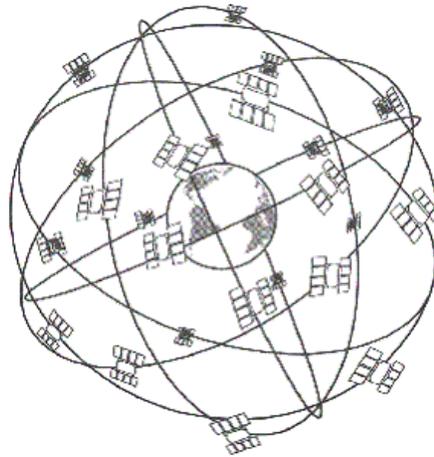
Dans la section 2, nous rappelons brièvement le problème multirobot de la constellation de satellites et nous présentons le modèle multiagent associé ainsi que le raisonnement de contrôle décentralisé proposé. Puis nous présentons dans la section 3 quels doivent être les critères d'évaluation et nous définissons des mesures abstraites sur ces critères ; ces mesures sont ensuite définies pour notre application dans la section 4. Nous présentons et discutons dans la section 5 les résultats d'évaluation de l'application.

---

1. *Scalability* en anglais.

## 2. Un système multisatellite

### 2.1. Description du problème



**Figure 1.** Une constellation de satellites

Nous nous intéressons au contrôle décentralisé d'un ensemble de satellites (généralement de 4 à 16) placés sur différentes orbites basses et dont la mission consiste à prendre des images de diverses zones de la surface terrestre. Le sol envoie aux satellites des tâches d'observation asynchrones et de priorités variables. Une durée minimum étant nécessaire pour réaliser une observation, deux tâches trop proches géographiquement ne peuvent pas être réalisées simultanément par le même satellite. De plus, certaines tâches sont décomposées en sous-tâches pouvant être réalisées par des satellites différents. Cette décomposition peut provenir du besoin de suivre l'évolution d'une même zone, de l'observer avec différents instruments ou de couvrir une zone géographique très étendue. Ces tâches décomposables sont appelées **tâches complexes**.

Les satellites doivent donc collaborer pour se répartir les tâches. Cette collaboration est soumise à trois contraintes :

- 1) comme prendre une image consomme des **ressources** de mémoire de masse, chaque satellite ne peut réaliser qu'un nombre limité de tâches avant de télécharger (c'est-à-dire d'envoyer les images correspondantes à une station au sol en visibilité) ;
- 2) les satellites doivent éviter les **redondances**, c'est-à-dire éviter que deux satellites distincts réalisent une même tâche alors qu'une seule réalisation aurait été suffisante : ceci permet d'économiser des ressources et de les utiliser pour réaliser d'autres tâches ;

3) lorsqu'une ou plusieurs sous-tâches d'une tâche complexe sont réalisées, toutes les autres sous-tâches de cette tâche complexe doivent l'être aussi.

Les constellations que nous considérons sont telles que les orbites des différents satellites se croisent régulièrement comme représenté sur la figure 1. Lors de ces rencontres, deux satellites (ou plus) peuvent communiquer entre eux *via* leur liaison intersatellite (ISL ou **I**nter**S**atellite **L**ink) sans intervention du sol. De plus, nous considérons des satellites autonomes en termes de planification, signifiant par-là qu'ils sont équipés d'algorithmes pour planifier des tâches de manière individuelle.

## 2.2. Modélisation d'une constellation de satellites autonomes

### 2.2.1. Le système multiagent

**Définition 1 (Constellation)** Une constellation  $\mathcal{S}$  est un triplet  $\langle \mathcal{A}, \mathbb{T}, \mathcal{V} \rangle$  tel que  $\mathcal{A} = \{a_1 \dots a_n\}$  est un ensemble de  $n$  agents représentant  $n$  satellites,  $\mathbb{T} \subset \mathbb{N}$  est un ensemble de dates définissant une horloge commune,  $\mathcal{V} : \mathcal{A} \times \mathbb{T} \mapsto 2^{\mathcal{A}}$  est une relation symétrique et non transitive qui spécifie pour un agent donné et une date donnée l'ensemble des agents avec lesquels il peut communiquer à cette date (modèle d'accointances).

La constellation  $\langle \mathcal{A}, \mathbb{T}, \mathcal{V} \rangle$  est une connaissance publique de chaque agent  $a_i \in \mathcal{A}$ . Comme  $\mathcal{V}$  est fondée sur les orbites des satellites et que ces dernières sont déterministes et périodiques, alors  $\mathcal{V}$  est périodique. Soit  $P \in \mathbb{T}$  la période de  $\mathcal{V}$  et  $\mathbb{T}_P \subset \mathbb{T}$  l'intervalle de temps de durée  $P$ .

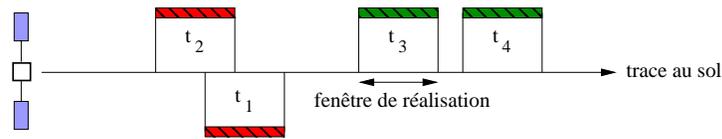
Chaque agent  $a_i$  possède aussi des connaissances privées à la date  $\tau$  : un sous-ensemble  $\mathcal{T}_{a_i}^\tau$  de tâches du système et un sous-ensemble  $\mathcal{I}_{a_i}^\tau$  de plans partiels (le sien compris) d'agents de  $\mathcal{A}$ .

### 2.2.2. Les tâches

**Définition 2 (Tâche)** Une tâche ou tâche atomique  $t$  est un tuple  $\langle ID_t, pos_t, prio(t), b_t \rangle$  tel que  $ID_t$  est l'identifiant unique de  $t$ ,  $pos_t$  représente les coordonnées géographiques de  $t$ ,  $prio(t) \in \mathbb{N}^*$  est une priorité et  $b_t \in \{\top, \perp\}$  est un booléen indiquant si  $t$  a été réalisée ou non.

À chaque tâche  $t$  que connaît un agent  $a_i$ , ce dernier associe une fenêtre de réalisation qui débute à la date où  $a_i$  peut réaliser  $t$  au plus tôt et se termine à une date fonction de la taille de  $t$ . Cette fenêtre est calculée à partir des coordonnées géographiques  $pos_t$  et de la position orbitale courante de  $a_i$ .

**Définition 3 (Réaliser une tâche atomique - observation)** Soit un agent  $a_i$  et une tâche  $t$  telle que  $b_t = \perp$ . Si  $a_i$  réalise  $t$  à la date  $\tau \in \mathbb{T}$ , noté  $a_i \vdash_\tau t$ , alors  $b_t = \top$ ,  $\forall \tau' > \tau$ . Dans toute la suite, nous appellerons observation une tâche atomique réalisée.



**Figure 2.** Exclusion mutuelle entre les tâches atomiques  $t_1$  et  $t_2$

### 2.2.3. Les contraintes sur les tâches

(1) Comme réaliser une tâche atomique nécessite du temps, deux réalisations de tâches atomiques ne peuvent pas être effectuées simultanément par un même satellite si leurs fenêtres de réalisation se chevauchent. Nous pouvons ainsi définir des **exclusions mutuelles** sur un ensemble de tâches atomiques. Par exemple sur la figure 2, les tâches  $t_1$  et  $t_2$  sont en exclusion mutuelle. Cette contrainte est une contrainte locale à un agent. En effet, les fenêtres de réalisation associées à une tâche atomique sont fonction de l'orbite de l'agent. Ainsi pour deux agents distincts, les exclusions mutuelles sur un même ensemble de tâches atomiques ne sont pas nécessairement les mêmes. De plus, il s'agit d'une contrainte dure (qui ne peut être violée) car elle est due à la mécanique spatiale et au mode d'utilisation des instruments de détection.

**Définition 4 (Exclusion mutuelle)** Soit un agent  $a_i \in \mathcal{A}$  et deux tâches atomiques  $t_1$  et  $t_2$ . On dit que  $t_1$  est en exclusion mutuelle avec  $t_2$  pour  $a_i$ , noté  $t_1 \bowtie t_2$ , si et seulement leurs fenêtres de réalisation pour  $a_i$  se chevauchent.

(2) Une **tâche complexe** est décomposée en tâches atomiques et chacune de ces tâches atomiques doit être réalisée pour que la tâche complexe le soit. Contrairement aux exclusions mutuelles, la réalisation d'une tâche complexe est une contrainte globale. Cependant, il s'agit d'une contrainte souple (qui peut être violée). Formellement :

**Définition 5 (Tâche complexe)** Une tâche complexe est un sous-ensemble  $\mathcal{T}$  de tâches atomiques.

**Définition 6 (Réaliser une tâche complexe)** Une tâche complexe  $\mathcal{T}$  est réalisée à la date  $\tau \in \mathbb{T}$  si et seulement si  $\forall t \in \mathcal{T}, b_t = \top, \forall \tau' > \tau$ .

(3) Lorsqu'une tâche atomique est réalisée, l'image générée est sauvegardée dans la mémoire de masse de l'agent. Cette ressource mémoire ne se régénère qu'au moment d'un téléchargement. Chaque agent étant limité en mémoire, éviter les redondances est souhaitable. Nous disons qu'il y a redondance lorsqu'une tâche atomique est réalisée par un agent alors qu'elle a déjà été réalisée par un autre agent. Éviter une redondance est une contrainte locale et souple. Formellement :

**Définition 7 (Redondance)** Soit  $a_i \in \mathcal{A}$  un agent tel que  $a_i \vdash_{\tau} t$ . Il y a redondance pour  $t$  si et seulement si  $\exists a_j \in \mathcal{A}$  et  $\exists \tau' \in \mathbb{T}$  ( $\tau' \leq \tau$ ) tels que  $a_j \vdash_{\tau'} t$ .

#### 2.2.4. Les plans partiels des agents

Les plans partiels sont représentés par des ensembles d'intentions. Une intention représente l'attitude d'un agent face à une tâche donnée. Nous pouvons la considérer comme une modalité de la proposition (**l'agent  $a_i$  réalise la tâche  $t$** ) :

**Définition 8 (Intention)** Une intention  $I_t^{a_i}$  de l'agent  $a_i$  envers la tâche  $t$  est un triplet  $\langle \text{mod}(I_t^{a_i}), \text{rea}(I_t^{a_i}), \text{tel}(I_t^{a_i}) \rangle$  tel que  $\text{mod}(I_t^{a_i}) \in \{\square, \diamond, \square\neg, \diamond\neg\}$  est la modalité de l'intention,  $\text{rea}(I_t^{a_i}) \in \mathbb{T} \cup \{\emptyset\}$  est la date de réalisation de la tâche  $t$  et  $\text{tel}(I_t^{a_i}) \in \mathbb{T} \cup \{\emptyset\}$  est la date de télédéchargement de l'image associée.

La sémantique des différentes modalités est la suivante :

- modalité  $\diamond$  : l'agent *propose* de réaliser une tâche mais cette intention est révisable ;
- modalité  $\square$  : l'agent *s'engage* à réaliser une tâche et il ne révisera pas son intention à moins qu'un autre agent s'engage à son tour ;
- modalité  $\diamond\neg$  : l'agent se *désiste faiblement*, c'est-à-dire qu'il ne réalisera pas la tâche, mais peut réviser cette intention ;
- modalité  $\square\neg$  : l'agent se *désiste fortement*, il ne réalisera pas la tâche et ne révisera pas son intention.

**Définition 9 (Plan)** Soit un agent  $a_i$  et  $\mathcal{T}_{a_i}^{\tau}$  l'ensemble des tâches atomiques connues par  $a_i$ .  $\forall t \in \mathcal{T}_{a_i}^{\tau}$ , le plan de l'agent  $a_i$  à la date  $\tau$  est défini ainsi :

- la tâche atomique  $t$  est planifiée  $\equiv (\text{mod}(I_t^{a_i}) \in \{\square, \diamond\})$  ;
- la tâche atomique  $t$  n'est pas planifiée  $\equiv (\text{mod}(I_t^{a_i}) \in \{\square\neg, \diamond\neg\})$ .

REMARQUE. — L'équivalence entre plan et ensemble d'intentions signifie que les dates de réalisation et de télédéchargement ne sont vides que si et seulement si l'intention associée possède une modalité de désistement (fort ou faible).

### 2.3. Méthode de contrôle décentralisé proposée

Nous avons proposé dans (Bonnet *et al.*, 2008, Bonnet *et al.*, 2007b) une méthode de coordination fondée sur un processus local à chaque agent :

1) Chaque agent  $a_i$  planifie individuellement en fonction des tâches  $\mathcal{T}_{a_i}^{\tau}$  qu'il connaît, de leur priorité ainsi que des contraintes de ressources et d'exclusions mutuelles. Le plan est un ensemble de **propositions** estampillées par leur date de génération qui peuvent être révisées lorsque l'agent reçoit de nouvelles tâches.

**Définition 10 (Planification)** Soit un agent  $a_i \in \mathcal{A}$  devant planifier à la date  $\tau \in \mathbb{T}$  un ensemble de tâches  $\mathcal{T}_{a_i}^\tau$ . Soit  $M \in \mathbb{N}$  le nombre maximum d'images que  $a_i$  peut enregistrer dans sa mémoire de masse. L'agent  $a_i$  planifie en résolvant le programme linéaire en nombres entiers suivant :

- a) soit  $\mathcal{T}^\perp = \{t_k \in \mathcal{T}_{a_i}^\tau : b_{t_k} = \perp\}$ ;
- b)  $\forall t_k \in \mathcal{T}^\perp$ , une variable  $v_k \in \{0, 1\}$  est générée;
- c)  $\forall t_k, t_{k'} \in \mathcal{T}^\perp$  en exclusion mutuelle,  $v_k + v_{k'} \leq 1$ ;
- d)  $\sum v_k \leq M$ .
- e)  $a_i$  cherche les  $\{v_k\}$  qui maximisent la fonction objectif<sup>2</sup> suivante :

$$\sum \frac{v_k}{prio(t_k)}$$

Après résolution :

- a)  $\forall v_k : v_k = 1$  et  $mod(\mathcal{I}_{t_k}^{a_i}) \notin \{\square, \square\bar{\neg}\}$ ,  $mod(\mathcal{I}_{t_k}^{a_i}) \leftarrow \diamond$
- b)  $\forall v_k : v_k = 0$  et  $mod(\mathcal{I}_{t_k}^{a_i}) \notin \{\square, \square\bar{\neg}\}$ ,  $mod(\mathcal{I}_{t_k}^{a_i}) \leftarrow \diamond\bar{\neg}$

2) Lorsque deux agents  $a_i$  et  $a_j$  se rencontrent, ils partagent leurs connaissances non seulement sur les tâches mais aussi sur leurs plans partiels et ceux des autres agents  $a_k$  afin de construire une connaissance commune. Ces informations sont donc propagées à l'intérieur de la constellation par un protocole épidémique (Gupta *et al.*, 2002, Jenkins *et al.*, 2001, Pittel, 1987) et la fonction  $\mathcal{V}$  permet de définir des **chaînes de communication** entre les agents. La figure 3 représente d'une part une communication directe entre les agents  $a_i$  et  $a_j$  et, d'autre part, une communication indirecte entre  $a_i$  et  $a_j$  par l'intermédiaire d'un agent  $a_k$ .

**Définition 11 (Communication)** Soit une constellation  $\langle \mathcal{A}, \mathbb{T}, \mathcal{V} \rangle$  et  $a_i, a_j \in \mathcal{A}$  deux agents.  $a_i$  communique avec  $a_j$  à  $(\tau_i, \tau_j) \in \mathbb{T}^2$  avec  $\tau_i$  la date d'émission de  $a_i$  et  $\tau_j$  la date de réception de  $a_j$  si et seulement si :

- communication directe :  $\exists \tau_i \in \mathbb{T}_P$  tel que  $a_j \in \mathcal{V}(a_i, \tau_i)$  (alors  $\tau_i = \tau_j$ ) OU
- communication indirecte :  $\exists l \in \mathbb{N}^*$  tel que  $\exists \{(a_{\tau_k}, \tau_k) \in \mathcal{A} \times \mathbb{T}, k \in [0 \dots l]\}$  avec  $a_{\tau_0} \in \mathcal{V}(a_i, \tau_i)$ ,  $a_{\tau_{k+1}} \in \mathcal{V}(a_{\tau_k}, \tau_k)$  ( $\tau_i < \tau_k < \tau_{k+1} < \tau_j$ ) et  $a_j \in \mathcal{V}(a_{\tau_l}, \tau_j)$ .

La figure 4 représente le graphe de communications entre trois agents  $a_1, a_2$  et  $a_3$ . Les sommets représentent les agents, les arêtes l'existence de communications épisodiques entre deux agents et les étiquettes les dates auxquelles ces communications ont lieu. Ici, l'agent  $a_1$  peut communiquer directement avec  $a_3$  aux dates (8, 8) et (16, 16) mais il peut aussi communiquer indirectement aux dates (2, 5) et (12, 15). La fonction  $\mathcal{V}$  étant une connaissance publique, chaque agent est en mesure de savoir à quels autres agents  $a_k$  une proposition de  $\mathcal{I}_\tau^{a_i}$  a été communiquée. Comme les agents sont homogènes dans leur raisonnement, ceci fonde la connaissance commune.

2. Dans le domaine spatial, plus la priorité est proche de 1, plus la tâche doit être réalisée impérativement.

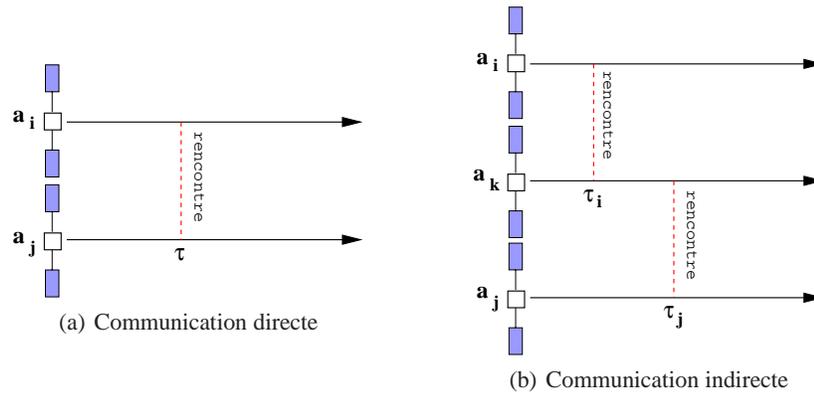


Figure 3. Représentation des deux types de communications

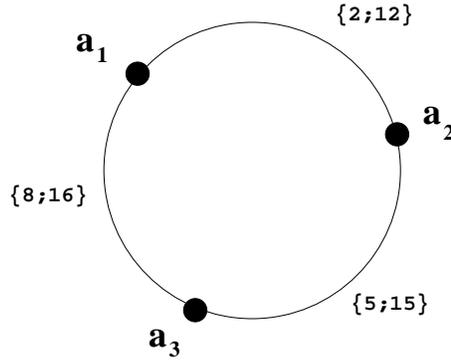


Figure 4. Exemple de graphe de communication

3) Chaque agent  $a_i$  calcule la **confiance** qu’il a en chacune des propositions qui lui ont été communiquées. Comme les agents n’ont pas de modèle de l’environnement, ils ne peuvent pas prédire l’arrivée de nouvelles tâches. Cependant, chaque rencontre est pour un agent une occasion de recevoir de nouvelles tâches et de réviser son plan partiel. En conséquence, une proposition reçue par  $a_i$  à la date  $\tau$  n’est plus nécessairement vraie à la date  $\tau'$  ( $\tau' > \tau$ ).

**Définition 12 (Confiance)** Soit  $a_i$  un agent qui connaît une proposition  $I_t^{a_j}$  de  $a_j$  à propos d’une tâche  $t$ .  $a_i$  calcule :

- la date de dernière confirmation de la proposition  $I_t^{a_j}$  pour  $a_i$  à la date  $\tau$  :

$$\tau^* = \max_{\substack{\tau_{K_{a_i}^\tau} < \tau_j \\ \tau_i < \tau}} \{ \tau_j : a_j \text{ communique } I_t^{a_j} \text{ à } a_i \text{ à } (\tau_j, \tau_i) \} \text{ et } I_t^{a_j} \text{ inchangée}$$

- le nombre d'agents que l'agent  $a_j$  rencontrera entre la date  $\tau^*$  et  $rea(I_t^{a_j})$  :

$$R_{\tau^*}^{a_i}(I_t^{a_j}) = \left| \bigcup_{\tau^* < \tau' < rea(I_t^{a_j})} \mathcal{V}(a_j, \tau') \right|$$

L'agent  $a_i$  a confiance dans l'agent  $a_j$  à propos de  $I_{t_j}^{a_j}$  si et seulement si

$$R_{\tau^*}^{a_i}(I_{t_j}^{a_j}) = 0$$

4) Chaque tâche complexe est représentée par une **coalition**. Les coalitions sont construites et manipulées *localement* par chaque agent, en fonction des connaissances qu'il a des autres agents *via* la communication. Ainsi, chaque agent utilise la notion de coalition pour raisonner et adapter ses propres intentions aux intentions des autres. Les coalitions sont donc formées implicitement au travers des intentions mais ne sont jamais construites explicitement par le système multiagent.

**Définition 13 (Coalition)** Une coalition  $C$  est un triplet  $\langle A, O, \mathfrak{P} \rangle$  tel que  $A \subseteq \mathcal{A}$  est un sous-ensemble d'agents membres de la coalition,  $O$  est un ensemble de tâches atomiques objectifs de la coalition et  $\mathfrak{P}$  un ensemble de tâches atomiques représentant le pouvoir de la coalition. Une coalition peut être dans différents états : complète si et seulement si  $O \subseteq \mathfrak{P}$ , minimale si et seulement si  $C$  est complète et  $A$  est minimal pour l'inclusion ( $\subseteq$ ).

**Définition 14 (Structure de coalition)** Chaque agent  $a_i$  génère la structure de coalitions courante de la manière suivante :

a)  $a_i$  partitionne l'ensemble des tâches atomiques  $\mathcal{T}_{a_i}^\tau$  en un ensemble  $\{\mathcal{T}_1 \dots \mathcal{T}_h\}$  tel que  $\mathcal{T}_i \subseteq \mathcal{T}_{a_i}^\tau$  est une tâche complexe ;

**Exemple 1** Supposons  $\mathcal{T}_{a_i}^\tau = \{t_1, t_2, t_3, t_4, t_5\}$ . Supposons que les tâches atomiques  $t_1$  et  $t_2$  soient les sous-tâches d'une tâche complexe  $\mathcal{T}_1$  et qu'il en est de même pour les tâches atomiques  $t_4, t_5$  et une tâche complexe  $\mathcal{T}_3$ . Alors  $\mathcal{T}_{a_i}^\tau$  est partitionné en  $\{\{t_1, t_2\}, \{t_3\}, \{t_4, t_5\}\}$ .

b) chaque  $\mathcal{T}_i$  est l'objectif d'une coalition potentielle. Comme ces ensembles sont disjoints, le nombre de coalitions générées par l'agent  $a_i$  dépend du nombre de tâches complexes qu'il connaît ;

c) du point de vue de  $a_i$ , les membres d'une coalition pour un sous-ensemble  $\mathcal{T}_i$  sont définis par :  $\{a_k \in \mathcal{A} : \exists t \in \mathcal{T}_i / \text{mod}(I_t^{a_k}) \in \{\square, \diamond\} \text{ et } a_i \text{ connaît } I_t^{a_k}\}$

**Exemple 2** Reprenons l'Exemple 1. Considérons la tâche  $t_3$  et supposons que  $\text{mod}(I_{t_3}^{a_i}) = \diamond$ ,  $\text{mod}(I_{t_3}^{a_k}) = \square$  et  $a_i$  connaît  $I_{t_3}^{a_k}$ .  $a_i$  peut construire la coalition  $C = \langle \{a_i, a_k\}, \{t_3\}, \{t_3\} \rangle$ . Cette coalition est complète mais pas minimale car  $\{a_i, a_k\}$  n'est pas minimal pour l'inclusion. Remarquons que  $a_i$  planifie  $t_3$  même s'il sait que  $a_k$  a fait de même. En effet, les intentions d'autrui ne sont pas prises en compte dans la planification : elles sont considérées lors des phases suivantes de la collaboration.

5) Chaque agent vérifie s'il est incité à rejoindre une coalition  $C$  pour accroître son pouvoir  $\mathfrak{P}$ . Les priorités des tâches dans  $O$  sont renforcées de manière à inciter

les agents à les planifier et ainsi de rejoindre la coalition  $C$ . Plus il y a de tâches déjà planifiées, plus les tâches restantes deviennent prioritaires.

**Définition 15 (Incitation)** Pour chaque coalition incomplète  $C$  d'objectif  $O$  et de pouvoir  $\mathfrak{P}$ , l'agent  $a_i$  exécute :

$$\forall t \in O, prio(t)' \leftarrow \frac{prio(t)}{1 + |\mathfrak{P}|}$$

6) Chaque agent  $a_i$  identifie et cherche à résoudre les redondances (appelées conflits) entre son plan et celui des autres agents. Cette étape consiste à transformer des propositions en **engagements**. Un engagement ne peut pas être révisé à moins qu'un autre agent s'engage à la place.

a) si le conflit est une connaissance commune des agents, l'agent (appelé **expert**) qui consommera le moins de ressources (c'est-à-dire qui conservera l'image dans la mémoire de masse le moins longtemps) s'engage à réaliser la tâche tandis que les autres la retirent de leur plan s'ils ont confiance en lui :

**Définition 16 (Minimisation 1)** Soit  $A^*$  l'ensemble des membres de la coalition avec lesquels l'agent  $a_i$  est en conflit sur une tâche  $t$  et  $A^+ \subseteq A^*$  les agents au courant du conflit. Soit  $rea^* = \min_{a_k \in A^+} rea(I_t^{a_k})$ . L'agent  $a_i$  applique :

(1) si  $a_i = \arg \min_{a_k \in A^+} |(rea(I_t^{a_k}) - rea^*, tel(I_t^{a_k}) - rea^*)|$  ( $a_i$  est l'agent expert) alors

$$mod(I_t^{a_i}) \leftarrow \square$$

(2) sinon soit  $a^*$  l'agent expert : si  $R_{\tau^*}^{a_i}(I_t^{a^*}) > 0$  alors  $mod(I_t^{a_i}) \leftarrow \diamond$  sinon  $mod(I_t^{a_i}) \leftarrow \square \neg$

b) si le conflit n'est pas une connaissance commune, les agents qui ignorent le conflit vont conserver leur proposition tandis que les autres la retirent de leur plan s'ils ont confiance dans les premiers :

**Définition 17 (Minimisation 2)** Soit  $A^*$  l'ensemble des membres de la coalition avec lesquels l'agent  $a_i$  est en conflit sur une tâche  $t$  et  $A^- \subseteq A^*$  les agents ignorant le conflit.  $a_i$  est au courant du conflit et applique :

(1) si  $\min_{a_k \in A^-} R_{\tau^*}^{a_i}(I_t^{a_k}) > 0$  alors  $mod(I_t^{a_i}) \leftarrow \diamond$

(2) sinon  $mod(I_t^{a_i}) \leftarrow \square \neg$

7) Chaque agent  $a_i$  relance un processus de planification individuelle (étape 1) en fonction des nouvelles contraintes, c'est-à-dire des engagements et du renforcement de priorité de certaines tâches.

**Définition 18 (Replanification)** L'agent  $a_i$  construit et résout le programme linéaire en nombres entiers de la définition 10 en y intégrant les contraintes suivantes :

a)  $\forall t_k \in \mathcal{T}^\perp, prio(t_k) = prio(t_k)'$  ;

b)  $\forall t_k \in \mathcal{T}^\perp$  telle que  $mod(\mathcal{I}_{t_k}^{a_i}) = \square, v_k = 1$  ;

c)  $\forall t_k \in \mathcal{T}^\perp$  telle que  $mod(\mathcal{I}_{t_k}^{a_i}) = \square \neg, v_k = 0$ .

Parce que cette méthode de contrôle est décentralisée dans un environnement dynamique, il n'est pas possible d'assurer qu'elle converge vers un optimum global. Cependant, elle ne dégrade pas les plans locaux des agents qui sont optimaux en fonction de leur connaissance. Ainsi, la méthode converge vers un optimum local. De plus, la méthode se termine. En effet, comme un engagement ( $\square$ ) d'un agent ne peut être révisé que si et seulement si un autre agent plus expert s'engage à sa place, il existe une date maximale à laquelle la tâche sera nécessairement réalisée.

#### 2.4. Premières évaluations

Un simulateur en Java a été implémenté. Il simule de manière centralisée le système décentralisé que forme la constellation à l'aide d'un système à événements discrets. Les propriétés du simulateur sont les suivantes :

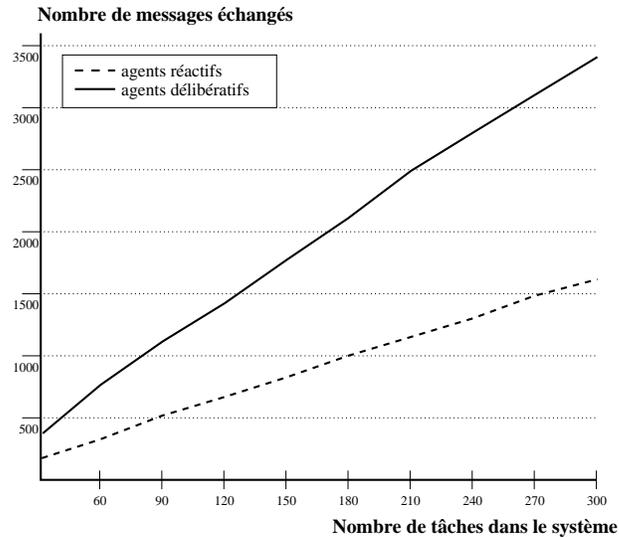
- 1) la machine sur laquelle les simulations ont été réalisées est une SunBlade 1 500 équipée d'un processeur Sparc v.9 cadencé à 1 062 MHz et d'une mémoire RAM de 1 Go ;
- 2) on dispose d'un modèle précis des orbites des satellites ;
- 3) les événements considérés sont les actions d'observation des agents, les communications des agents avec le sol (arrivées asynchrones de nouvelles tâches et télé-déchargements) et les communications intersatellites ;
- 4) il n'y a pas de bruit sur les communications ;
- 5) les événements simultanés correspondent à une concurrence dans l'exécution des actions d'observation ou des communications des agents. Dans ces cas, l'agent ayant l'identifiant le plus petit a la priorité et ses événements associés sont hiérarchisés : observations, communications avec le sol et communications intersatellites. Cette méthode, similaire à celle de (Chow, 1996), permet de rejouer une simulation de manière déterministe ;
- 6) un observateur central collecte les données locales des agents afin d'obtenir des mesures de performance.

Les agents satellites que nous considérons sont donc des **agents délibératifs** qui négocient les tâches à réaliser et se coordonnent à l'aide du mécanisme précédemment présenté. Afin d'évaluer notre approche, nous proposons de la comparer à deux autres protocoles décentralisés plus simples :

- 1) des **agents isolés** qui ne communiquent pas et ne se coordonnent pas : il s'agit d'un système *témoin* ;
- 2) des **agents réactifs** qui ne communiquent que des informations sur les tâches et ne raisonnent pas sur les plans partiels et les coalitions. Ces agents sont réactifs car leur seule stratégie est de retirer les tâches déjà réalisées de leur plan.

Afin de souligner les deux différences majeures (en dehors des capacités de calcul nécessaires) entre agents réactifs et délibératifs, un premier jeu de simulations a été

réalisé sur une constellation comportant 3 satellites. Toutes les 6 heures (en temps simulé), la station au sol envoie 40 nouvelles tâches complexes (composées d'au moins 2 tâches atomiques) à l'un des trois satellites. Différentes métriques nous permettent d'évaluer les résultats : le nombre de messages échangés en fonction du nombre de tâches (figure 5), le nombre de tâches réalisées (figure 6 et figure 8) et le nombre de tâches réalisées sans redondances (figure 7 et figure 9). Les figures 8 et 9 illustrent le gain des agents délibératifs et des agents réactifs relativement aux résultats des agents isolés.



**Figure 5.** Messages échangés en fonction du nombre de tâches

Notons que, dans ces expériences, les agents ne sont pas limités en ressources mémoire, c'est-à-dire qu'ils peuvent réaliser autant d'observations que possible entre deux télédownloads étant donné les contraintes d'exclusion mutuelle. Lors d'une mission d'observation réelle, les agents sont contraints en ressources mémoire et le nombre de tâches réalisées sans redondances est la métrique la plus importante. Nous pouvons alors remarquer que :

1) les agents réactifs échantent un nombre de messages inférieur aux agents délibératifs car les plans partiels ne sont pas communiqués ;

2) les agents délibératifs permettent de réaliser plus de tâches et de diminuer le nombre de redondances. Au fur et à mesure que le temps passe, l'écart entre les agents délibératifs et les agents réactifs se creuse : les agents délibératifs économisent des ressources et pourront les réaffecter plus tard pour réaliser plus de tâches. Ainsi les agents réactifs permettent d'atteindre un gain d'environ 20 % en tâches réalisées et 7,5 % en tâches réalisées sans redondances tandis que les agents délibératifs atteignent un gain de 30 % en tâches réalisées et 50 % en tâches réalisées sans redondances.

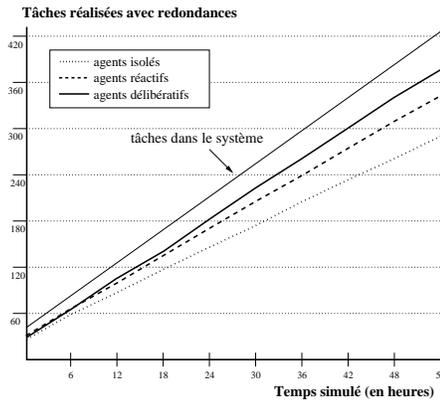


Figure 6. Réalisations

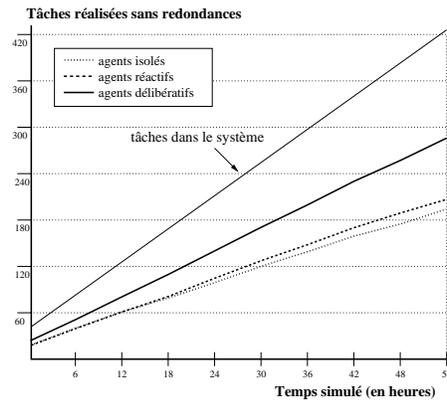


Figure 7. Réalisations sans redondances

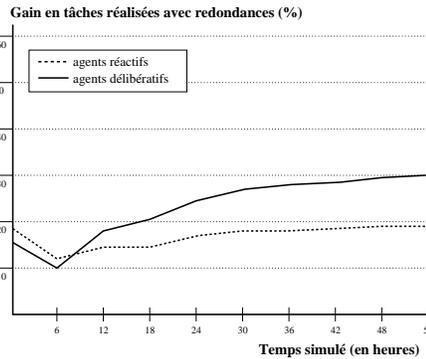


Figure 8. Gain (avec redondances)

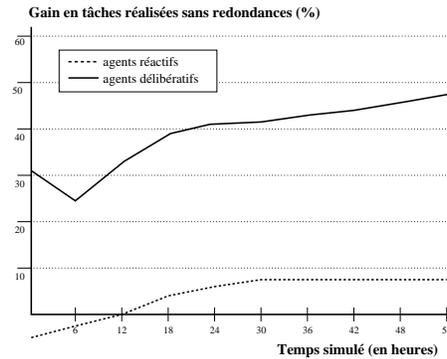


Figure 9. Gain (sans redondances)

L'étape suivante de l'évaluation consiste à « passer à l'échelle », c'est-à-dire à considérer une constellation composée d'un nombre plus élevé de satellites. Cependant pour un problème donné, augmenter le nombre de satellites – même isolés – permet de résoudre plus facilement le problème. En effet, de par la structure périodique du problème, un unique agent peut le résoudre, mais en un laps de temps important ; l'ajout d'autres agents en accélère la résolution. En soi le nombre de satellites augmente les performances mais trois questions se posent alors :

1) quelles sont les mesures pertinentes pour évaluer la performance d'un tel système ?

2) comment prendre en compte le biais dû à la diminution de la complexité d'un problème au fur et à mesure que le nombre d'agents affectés à sa résolution augmente ?

3) à partir de ces deux notions, peut-on définir une relation entre le nombre de satellites mis en œuvre et leur mode de coordination ?

### 3. Évaluer un système multiagent

Dans notre contexte, il ne s'agit pas d'évaluer la méthodologie qui mène à la construction d'un système multiagent (Bernon *et al.*, 2003, Sturm *et al.*, 2003, Zambonelli *et al.*, 2003) mais d'évaluer le système multiagent en fonctionnement comme le font par exemple (Contet *et al.*, 2008) pour les agents réactifs. (Lee *et al.*, 1998) proposent de le faire selon les trois axes classiques pour évaluer les systèmes distribués : performance, stabilité et capacité d'extension.

#### 3.1. Trois critères

La *performance* du système se mesure par le biais d'indicateurs statistiques comme la consommation en ressources, le temps de réponse, le nombre de tâches réalisées, le temps de calcul et la charge de communication. (Davidsson *et al.*, 2003) définissent par exemple les indicateurs suivants : réactivité du système, répartition des tâches entre les agents, utilisation de toutes les ressources, charge de communication. (Dutta *et al.*, 2002) proposent de n'utiliser que le temps d'exécution et la qualité d'exécution.

La *stabilité* est liée au concept d'équilibre. Lorsqu'un événement vient influencer sur le système, ce dernier est stable s'il parvient à revenir à un état d'équilibre. C'est un concept à rapprocher de la robustesse fonctionnelle du système (Davidsson *et al.*, 2003). Intuitivement, un système multiagent est à l'équilibre si ses performances restent stationnaires suite à un ou plusieurs événements qui peuvent venir le perturber (Lee *et al.*, 1998). La notion d'événement doit bien évidemment être définie : il peut s'agir de l'ajout ou du retrait d'agents, de l'ajout de nouvelles tâches, de bruit sur les observations ou de toute autre variation de paramètres au cours de l'exécution.

Si la définition des deux critères précédents est communément admise, ce n'est pas le cas pour la *capacité d'extension* dont la définition est liée à la complexité du système, terme difficile à définir pour les systèmes multiagents (Kubera *et al.*, 2007). Par exemple, (Lee *et al.*, 1998) s'intéressent à la complexité algorithmique et définissent la capacité d'extension comme le fait que la performance au pire cas du système est bornée par une fonction polynomiale de la charge de travail. De leur côté, (Turner *et al.*, 2000) considèrent un système comme extensible si la consommation en ressources par agent est bornée par une fonction polynomiale de la taille du système. Mais pour (Davidsson *et al.*, 2003), la capacité d'extension correspond à la dégradation (ou non) de la performance à mesure que le système croît en nombre d'agents.

La première définition revient en fait à une mesure de performance et ne s'intéresse pas à la dimension multiagent du système. La seconde définition prend en compte cette dimension mais définit un système comme extensible si les agents utilisent individuellement de moins en moins de ressources pour résoudre leur problème. Or dans le domaine qui nous intéresse, il y a une contradiction : si des ressources sont disponibles, elles doivent être utilisées. La troisième définition est plus pertinente pour notre application car la capacité d'extension réfère à la capacité du système à accroître ses performances alors que sa taille augmente elle aussi. Par exemple, quatre agents sont-ils deux fois plus efficaces que deux agents ?

Nous proposons de définir des mesures sur les notions de performance, stabilité et capacité d'extension pour un système multiagent physique mis en œuvre ou simulé. Dans les deux cas, nous parlerons d'expérience, que nous définissons formellement.

**Définition 19 (Expérience)** Une expérience  $\mathcal{E}$  est un tuple  $\langle E, \mathbb{M}, S, \mathbb{T} \rangle$  où  $E = \langle e_i \rangle$  est un ensemble de  $n$  entrées et  $S = \langle s_j \rangle$  est un ensemble de  $m$  sorties où les domaines de valeurs de chacun des éléments de  $E$  et  $S$  sont quelconques ;  $\mathbb{T} \subseteq \mathbb{N}$  est une horloge et  $\mathbb{M} : E \times \mathbb{T} \mapsto S$  est un modèle de la dynamique qui lie  $S$  à  $E$ .

$E$  est l'ensemble des paramètres initiaux de l'expérience (contraintes sur les objectifs des agents, nombre de tâches connues, nombre d'agents, ressources, conditions d'arrêt de la simulation, etc.) et  $S$  l'ensemble des observations en sortie. La fonction  $M$  est considérée comme une *boîte noire* qui modélise le système multiagent ainsi que l'environnement. Notons que nous considérons un *temps discrétisé*.

### 3.2. Performance

Une expérience produit une grande quantité de données dont le résultat dépend de nombreux paramètres (comme le nombre d'agents, les différentes échelles de complexité du problème, les paramètres intrinsèques des agents ou des interactions). Ainsi, il n'est pas trivial de définir de *bonnes mesures* pour évaluer les performances du système. Nous proposons de définir formellement ce qu'est une performance.

**Définition 20 (Performance)** Soit une expérience  $\mathcal{E}$ , un ensemble  $V = \{v_i \in E \cup S \cup \mathbb{T} : \exists v_i \in S\}$ . Une performance  $G(\mathcal{E})$  sur cette expérience est définie par :

$$G(\mathcal{E}) : V \mapsto \mathbb{R}$$

Cette définition nous permet de considérer  $G(\mathcal{E})$  comme une fonction quelconque des variables du système. Cependant nous nous limitons aux mesures de performance monovaluées, c'est-à-dire monocritères. La raison tient au fait que nous définissons la stabilité et la capacité d'extension en fonction des performances et que, dans un cadre multicritère, un système peut être stable (ou extensible) pour un critère et pas l'autre. Ceci nous amène à analyser chaque critère, c'est-à-dire chaque performance, séparément.

### 3.3. Stabilité et capacité d'extension

Trouver des mesures pertinentes lors d'un changement d'échelle est un problème que les sciences physiques traitent depuis longtemps à travers le concept de *grandeur sans dimension*. Ces grandeurs servent à étudier les propriétés intrinsèques des modèles. Le nombre de Reynolds, par exemple, représente le rapport entre les forces d'inertie et les forces visqueuses en dynamique des fluides. Un autre exemple est le nombre de Mach qui exprime le rapport de la vitesse locale d'un fluide sur la vitesse du son dans ce même fluide.

De manière générale, une grandeur sans dimension est une quantité permettant de décrire une caractéristique physique sans unité explicite d'expression. Elle est constituée du produit ou rapport de grandeurs ayant une dimension, de telle façon que le rapport des unités soit 1 (Giacomo, 1996).

Nous pouvons définir la stabilité et la capacité d'extension comme des grandeurs sans dimension sur les mesures de performance.

#### 3.3.1. Stabilité

Suite à un événement, nous supposons que les performances du système vont se dégrader. La stabilité correspond donc à la robustesse du système à cette dégradation de performance. Un événement est défini de la manière suivante :

**Définition 21 (Événement)** Soit une expérience  $\mathcal{E} = \langle E, \mathbb{M}, S, \mathbb{T} \rangle$ . Le système modélisé par  $\mathbb{M}$  est soumis à un événement  $\delta$  à la date  $\tau \in \mathbb{T}$  si  $E(\tau + 1) \neq E(\tau)$ . Notons  $\mathcal{E}^\delta$  l'expérience fondée sur  $\mathbb{M}$  et subissant l'événement  $\delta$  à la date  $\tau$ .

Dans ce cas, la stabilité est définie comme suit :

**Définition 22 (Stabilité)** Soit une expérience  $\mathcal{E} = \langle E, \mathbb{M}, S, \mathbb{T} \rangle$  et soit  $\mathcal{E}^\delta$  l'expérience fondée sur  $M$  subissant l'événement  $\delta$  à la date  $\tau \in \mathbb{T}$ . Soit  $G(\mathcal{E}(\tau'))$  une performance à la date  $\tau' > \tau$ . La stabilité du système est donnée par :

$$\mathcal{S}(\mathcal{E}, \mathcal{E}^\delta, \tau') = \frac{G(\mathcal{E}^\delta(\tau'))}{G(\mathcal{E}(\tau'))}$$

#### 3.3.2. Capacité d'extension

En considérant la capacité d'extension comme la variation des performances du système à mesure que le nombre d'agents augmente, la capacité d'extension est la variation du rapport de deux performances pour un nombre d'agents différent *ceteris paribus*.

**Définition 23 (Capacité d'extension)** Soit une expérience  $\mathcal{E}_n$  portant sur  $n$  agents et une performance  $G(\mathcal{E}_n)$ . La capacité d'extension du système à  $(n+p)$  agents ( $p \neq 0$ ) est donnée par :

$$X(n, p) = \frac{\binom{G(\mathcal{E}_{n+p})}{n+p}}{\binom{G(\mathcal{E}_n)}{n}}$$

Une capacité d'extension valant 1 indique que la performance par agent reste inchangée si le nombre d'agents augmente. Si la capacité d'extension est inférieure à 1 cela signifie que les agents ne sont plus aussi performants individuellement (même si le système dans sa globalité peut être plus performant). Une capacité d'extension supérieure à 1 indique qu'augmenter le nombre d'agents permet d'accroître la performance par agent et, par conséquent, d'accroître la performance du système en plus du gain dû au simple nombre d'agents.

Ainsi si une mesure de performance  $G(\mathcal{E}_n)$  est sous-linéaire<sup>3</sup> en fonction du nombre d'agents, le système n'est pas extensible. Si la mesure de performance est super-linéaire en fonction du nombre d'agents, le système est extensible.

#### 4. Application

En considérant les caractéristiques de la constellation de satellites présentée dans la section 2, les définitions abstraites de performance sont instanciées et nous adaptons les définitions de stabilité et capacité d'extension. De plus, nous proposons une mesure de coordination pour mettre en perspective le compromis entre nombre de satellites et capacités de raisonnement (réactifs ou délibératifs).

##### 4.1. Performances

Nous définissons trois performances : la *réactivité*, c'est-à-dire la capacité de  $n$  agents à accomplir plus rapidement qu'un unique agent un ensemble de tâches, l'*efficacité*, c'est-à-dire une mesure des redondances au sein d'un ensemble de tâches et l'*activité*, la proportion de tâches réalisées à une date donnée.

**Définition 24 (Réactivité)** Soit une expérience  $\mathcal{E}$  et  $\mathbb{T}_n \in S$  le temps nécessaire à  $n$  satellites pour réaliser toutes les tâches. La réactivité de la constellation est donnée par :

$$\frac{\mathbb{T}_1}{\mathbb{T}_n}$$

La mesure de réactivité est toujours supérieure à 1 car  $n$  satellites ( $n > 1$ ) ne peuvent pas avoir besoin de plus de temps qu'un unique satellite pour réaliser un

3. Une fonction  $f(n)$  est sous-linéaire si  $f(n) \in o(n)$ , c'est-à-dire  $\lim_{n \rightarrow \infty} \frac{f(n)}{n} = 0$ .

ensemble de tâches. Ainsi plus la réactivité est élevée, meilleure est la performance de la constellation.

**Définition 25 (Efficacité)** Soit une expérience  $\mathcal{E}$ ,  $O \subset S$  l'ensemble des observations réalisées et  $T \subset S$  l'ensemble des tâches réalisées. L'efficacité de la constellation est donnée par :

$$\frac{|T|}{|O|}$$

Dans notre application, le nombre d'observations  $|O|$  correspond au nombre d'images prises par l'ensemble des satellites, en incluant les éventuelles redondances. Le nombre de tâches  $|T|$  correspond au nombre de tâches réalisées par la constellation, sans tenir compte des redondances. Ainsi le nombre d'observations est toujours supérieur ou égal au nombre de tâches réalisées. La mesure d'efficacité est donc toujours inférieure ou égale à 1. Plus elle est élevée, meilleure est la performance de la constellation.

**Définition 26 (Activité)** Soit une expérience  $\mathcal{E}$ ,  $T \subset E$  l'ensemble des tâches dans le système et  $T_\tau \subset S$  l'ensemble des tâches réalisées à la date  $\tau \in \mathbb{T}$ . L'activité du système multiagent est :

$$A(\mathcal{E}, \tau) = \frac{|T_\tau|}{|T|}$$

Le nombre de tâches  $|T_\tau|$  correspond au nombre de tâches dans le système réalisées par la constellation à la date  $\tau$  sans tenir compte du nombre de redondances. Ainsi l'activité indique la proportion de tâches réalisées à la date  $\tau$ . L'activité est donc toujours inférieure ou égale à 1. Plus elle est élevée, plus la constellation est performante à la date courante.

## 4.2. Stabilité

Les événements que nous considérons sont l'arrivée de nouvelles tâches dans le système.

**Définition 27 (Flot de tâches)** Soit une expérience  $\mathcal{E} = \langle E, \mathbb{M}, S, \mathbb{T} \rangle$  où  $\mathcal{T}(\tau) \subset E$  est l'ensemble des tâches dans le système à la date  $\tau$ . Il y a un flot de tâches  $\delta$  à la date  $\tau \in \mathbb{T}$  si  $|\mathcal{T}(\tau + 1)| > |\mathcal{T}(\tau)|$ . Notons  $\mathcal{E}^\delta$  cette expérience recevant le flot  $\delta$ .

Nous définissons alors la stabilité sur l'activité.

**Définition 28 (Stabilité face à la charge de travail)** Soit une expérience  $\mathcal{E} = \langle E, \mathbb{M}, S, \mathbb{T} \rangle$  et soit  $\mathcal{E}^\delta$  l'expérience fondée sur  $\mathbb{M}$  recevant un flot de tâches  $\delta$  à la

date  $\tau \in \mathbb{T}$ . Soit  $A(\mathcal{E}, \tau)$  l'activité à la date  $\tau$ . La stabilité du système face à la charge de travail est donnée par :

$$S(\mathcal{E}, \mathcal{E}^\delta) = \frac{A(\mathcal{E}^\delta, \tau + 1)}{A(\mathcal{E}, \tau + 1)}$$

La stabilité représente la dégradation de l'activité de la constellation lorsque la charge de travail (c'est-à-dire le nombre de tâches à réaliser) augmente. Cette mesure est toujours inférieure à 1. Plus elle est élevée, meilleure est la robustesse de la constellation à une modification de la charge de travail. Ainsi, une valeur de 1 signifie que l'événement n'a aucune influence sur l'activité de la constellation.

### 4.3. Capacité d'extension

Dans notre application, nous nous intéressons à des constellations de taille 1, 4, 6, 8, 9, 12 et 16 respectivement. La capacité d'extension pour la réactivité et l'efficacité exprime alors l'intérêt à passer d'une taille de constellation à la suivante, comme par exemple de 1 à 4 satellites ou de 9 à 12 satellites. Dans notre cas, cette mesure reste inchangée par rapport à la définition 23.

### 4.4. Capacité de coopération

Comme nous nous intéressons à deux types d'agents,

- 1) des agents réactifs qui évitent de réaliser les tâches qui l'ont déjà été ;
- 2) des agents délibératifs qui négocient les tâches à réaliser et construisent des coalitions,

nous proposons une mesure pour mettre en évidence le gain à utiliser des agents délibératifs plutôt que des agents réactifs en dépit d'un coût de développement et de communication plus élevé. Formellement,

**Définition 29 (Capacité de coopération)** Soit une expérience  $\mathcal{E}_k$  impliquant  $k$  agents réactifs et une performance  $G$ . Soit une expérience  $\mathcal{E}_n$  identique si ce n'est qu'elle implique  $n$  agents délibératifs à la place. La capacité de coopération de la constellation est donnée par :

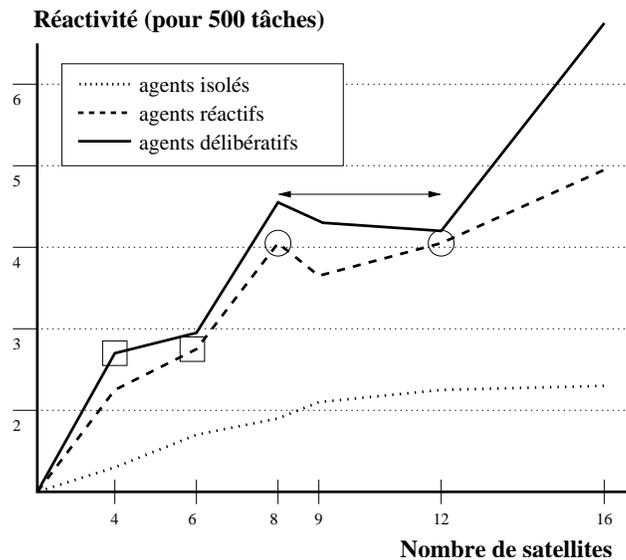
$$C(\mathcal{E}_k) = \min\{n : G(\mathcal{E}_n) \geq G(\mathcal{E}_k)\}$$

La capacité de coopération indique combien d'agents délibératifs suffisent pour être aussi performants qu'une constellation composée de  $k$  agents réactifs. Plus cette mesure est faible, plus la taille de la constellation peut être réduite pour une performance identique.

## 5. Expériences et résultats

Nous considérons alors des expériences sur un scénario comportant 500 tâches initiales et un satellite unique ou des constellations de (Walker, 1984) de tailles différentes (4, 6, 8, 9, 12 ou 16 satellites régulièrement répartis sur un nombre fini de plans orbitaux). Une expérience se termine lorsque toutes les tâches ont été réalisées. Les performances sont ensuite mesurées.

### 5.1. Réactivité de la constellation



**Figure 10.** Réactivité de la constellation

Sur la figure 10, nous pouvons remarquer que la réactivité des agents isolés suit une courbe logarithmique tandis que les agents réactifs et délibératifs ne présentent pas une telle régularité. Cela est dû à la structure hétérogène des interactions entre les satellites. En effet, des agents isolés sont toujours homogènes dans le sens où il n'y a aucune interaction entre eux. En revanche, lorsqu'il y a interaction pour les agents réactifs et délibératifs, celle-ci ne peut exister qu'entre des satellites placés sur des plans orbitaux différents. La figure 1 représente quelques-uns de ces plans.

Ainsi, 4 plans de 2 satellites peuvent permettre plus d'interactions que 3 plans de 4 satellites. Il y a donc une notion de topologie des interactions que l'on ne trouve pas dans d'autres applications multirobots comme le fourragement (Shell *et al.*, 2006, Parker, 1998). En fait le nombre de satellites n'est pas le facteur déterminant dans le

nombre d'interactions, il s'agit plutôt de leurs orbites : peu de satellites peuvent communiquer souvent tandis que de nombreux satellites peuvent ne communiquer qu'épisodiquement. Cette topologie des interactions provoque une réduction de la performance de la constellation. Ce phénomène peut être observé dans la plage de 8 à 12 agents délibératifs. Sur cette plage, représentée par la double flèche de la figure 10, il y a une perte de réactivité lorsque l'on accroît le nombre de satellites de 8 à 12.

Le tableau 1 nous donne la capacité d'extension des agents délibératifs pour le scénario précédent selon le critère de réactivité. Nous pouvons remarquer que passer de 6 à 8 satellites, tout comme passer de 12 à 16 satellites, permet d'accroître la réactivité au-delà de ce que le simple nombre d'agents pouvait procurer. Ainsi mettre en œuvre 8 satellites comparativement à 6 permet non seulement d'obtenir le gain de performance attendu par rapport au nombre, mais d'obtenir une réactivité accrue de 16 %. De même, passer de 12 à 16 satellites permet d'accroître la réactivité de 26 % en plus du gain dû au nombre.

Cependant nous avons des cas sous-linéaires, c'est-à-dire des cas où le système n'est pas extensible. Ceci est dû, une nouvelle fois, à la topologie des interactions entre les agents. En effet, si ajouter un nouvel agent n'augmente pas proportionnellement le nombre d'interactions positives au sein du système, ce dernier ne va pas accroître suffisamment sa réactivité pour être considéré comme extensible.

Nb. de satellites délibératifs		$X(n, p)$
$n$	$n + p$	
1	4	0,67
4	6	0,73
6	8	<b>1,16</b>
8	9	0,85
9	12	0,74
12	16	<b>1,26</b>

**Tableau 1.** Capacité d'extension (réactivité)

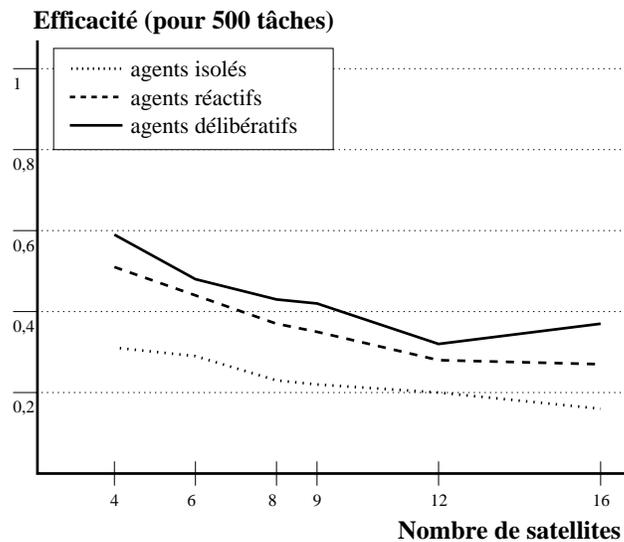
Le tableau 2 indique la capacité de coopération du système selon le critère de réactivité. Les valeurs ont été arrondies à l'entier le plus proche afin de prendre en compte la granularité de la constellation de Walker (c'est-à-dire  $n$  satellites et  $m$  plans orbitaux de  $k$  tels que  $n = k \times m$ ). Nous pouvons remarquer que 4 agents délibératifs suffisent pour être aussi réactifs que 6 agents réactifs ( $\square$  sur la figure 10). Il y a une économie au niveau de la mise en œuvre de la constellation. De même, 8 agents délibératifs sont au moins aussi performants que 12 agents réactifs ( $\circ$  sur la figure 10). Cependant, nous pouvons remarquer que 8 agents réactifs sont aussi performants que 12 agents réactifs, ce qui nous a conduits à ne considérer que le cas d'équivalence en termes de capacité de coopération entre 8 agents délibératifs et 8 agents réactifs. Deux autres cas d'équivalence, sur le tableau 2, pour  $k = 4$  et 16 peuvent être mis en évidence. Il

n'existe pas un nombre inférieur d'agents délibératifs au moins aussi performants que  $k$  agents réactifs. Dans ces cas, une question doit être posée : le gain en performance procuré par les agents délibératifs contrebalance-t-il le coût de développement de tels satellites ?

Nb. de satellites $k$	$C(\mathcal{E}_k)$
4	4
6	4
8	8
9	8
12	8
16	16

**Tableau 2.** Capacité de coopération (réactivité)

### 5.2. Efficacité de la constellation



**Figure 11.** Efficacité de la constellation

La figure 11 nous permet de remarquer que les agents délibératifs sont en moyenne plus efficace au sens de la définition 25 de 5 % que les agents réactifs et de 19 % que les agents isolés. Ceci correspond à une économie en termes de mémoire de masse utilisée pour stocker les images. Si la mesure d'efficacité décroît au fur et à mesure que le

nombre de satellites augmente, cela signifie que le système n'est pas extensible selon notre définition (se reporter au tableau 3). En effet, le gain en termes de ressources n'est pas proportionnel au nombre d'agents. Nous pouvons cependant remarquer que le système est extensible au sens de Turner (Turner *et al.*, 2000) : un système est extensible si la consommation en ressources est bornée par une fonction polynomiale du nombre d'agents. Dans notre application, la consommation en ressources peut être mesurée par le nombre d'observations divisé par le nombre de tâches réalisées  $\frac{|O|}{|T|}$ , c'est-à-dire quelle quantité de mémoire de masse discrétisée en images est nécessaire par tâche. Or, c'est l'inverse de la mesure d'efficacité. Ainsi, la consommation en ressources peut être bornée par une fonction polynomiale en le nombre d'agents.

Nb. de satellites délibératifs		X(n, p)
n	n + p	
1	4	0,15
4	6	0,19
6	8	0,15
8	9	0,16
9	12	0,1
12	16	0,08

**Tableau 3.** Capacité d'extension (efficacité)

Étant donné que l'efficacité est décroissante à mesure que le nombre d'agents s'accroît, un nombre minimal d'agents délibératifs sera toujours plus efficace qu'un nombre quelconque d'agents réactifs. Ainsi les agents délibératifs sont toujours intéressants pour l'efficacité de la constellation. Cependant, il convient de se poser la question suivante : le gain d'efficacité de 5 % procuré par les agents délibératifs compense-t-il le coût de développement de tels satellites ?

### 5.3. Stabilité de la constellation

Le stabilité est ici mesurée à partir d'un scénario comportant une constellation de Walker et 250 tâches initiales. Après 6 heures (temps simulé), un événement ajoute 250 tâches puis la stabilité est mesurée 6 heures après, comme le représente la figure 12.

Nous pouvons remarquer deux phénomènes sur la figure 13 :

- 1) la stabilité croît dans un premier temps puis devient constante ;
- 2) les agents délibératifs et réactifs sont aussi stables les uns que les autres.

Le premier résultat vient du fait que la stabilité dépend en partie de la capacité des agents à construire une connaissance commune en termes de tâches à réaliser et

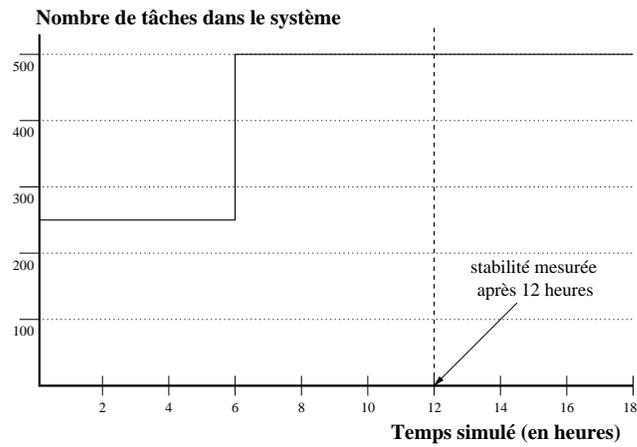


Figure 12. Évolution de la charge de travail de la constellation

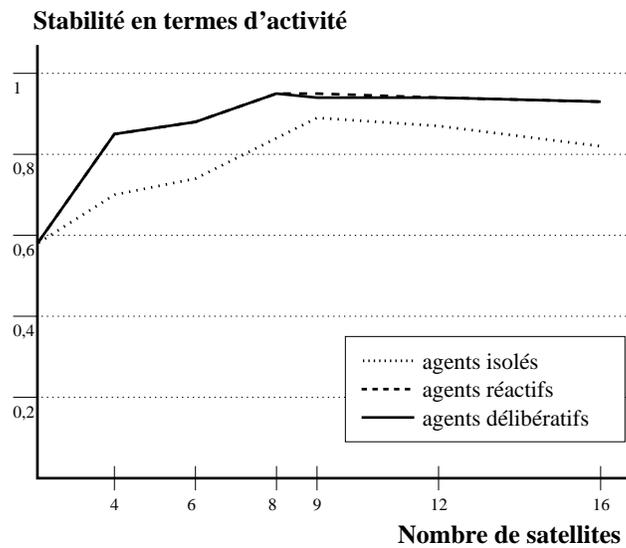


Figure 13. Stabilité de la constellation

de tâches déjà réalisées. Ajouter des agents permet de transmettre plus rapidement de l'information et de bâtir plus vite cette connaissance commune. Cependant au-delà d'un certain nombre d'agents, ajouter un nouvel agent n'a pas d'influence sur la vitesse de ce processus.

Le deuxième résultat provient de deux phénomènes qui interagissent. D'un côté, les agents délibératifs sont plus efficaces (voir section 5.2) et, parce qu'ils économisent des ressources pouvant être réaffectées par la suite, ils peuvent absorber la charge de travail supplémentaire. D'un autre côté, les agents délibératifs prennent des engagements qui peuvent devenir sous-optimaux lorsque de nouvelles tâches arrivent et ces engagements ne peuvent pas être révisés à moins qu'un autre agent ne s'engage à son tour. Ainsi, la performance des agents délibératifs est contrebalancée par leur sensibilité aux nouveaux événements.

Nb. de satellites délibératifs		$X(n, p)$
$n$	$n + p$	
1	4	0,37
4	6	<b>1,02</b>
6	8	0,81
8	9	0,95
9	12	0,75
12	16	0,74

**Tableau 4.** Capacité d'extension (stabilité)

Le tableau 4 indique la capacité d'extension du système en termes de stabilité. Passer de 4 à 6 satellites permet de légèrement accroître la stabilité au-delà de ce qui est attendu. La capacité d'extension décroît par la suite, atteignant une valeur maximale comme présenté sur la figure 13.

Comme la mesure de stabilité est identique pour les agents délibératifs et réactifs, la mesure de coopération est égale au nombre de satellites considérés. Il n'y a pas de gain à utiliser des agents délibératifs en ce qui concerne la stabilité du système car les agents réactifs sont naturellement stables en raison de leur coût d'organisation nul au sein de la constellation. Or comme il y a équivalence en termes de stabilité entre agents délibératifs et réactifs, nous pouvons dire que le coût d'organisation des agents délibératifs est compensé par le gain qu'ils permettent.

## 6. Conclusion

Évaluer un système multirobot en fonctionnement nécessite de définir des mesures pertinentes qui permettent la comparaison de différentes simulations ou expériences dans un cadre réel. Nous avons proposé une manière de formaliser des mesures sur des notions de performance, de stabilité et capacité d'extension.

Ces mesures sont mises en pratique afin d'évaluer un algorithme de coordination décentralisé dans le domaine des satellites d'observation. La simulation d'une constellation de satellites nous a permis de dégager des mesures de *réactivité*, d'*efficacité* et

de *stabilité* afin de mettre en évidence le compromis nécessaire entre nombre d'agents du système et extension des performances. Nous avons pu nous rendre compte que les notions de *capacité d'extension* ne suffisaient pas à comparer différentes approches et nous avons défini une *capacité de coopération* pour mettre en évidence le gain à utiliser des algorithmes de coordination plus complexes. De plus, nous pouvons remarquer d'une part que capacité d'extension et capacité de coopération sont des mesures relatives à un critère de performance. Un système multiagent n'est donc pas extensible en soi mais selon un critère donné. Par ailleurs, plusieurs critères peuvent être contradictoires.

Par exemple dans notre application, si on désire un système stable en termes d'activité, on préférera une constellation ayant un faible nombre d'agents réactifs ; en revanche si on privilégie la réactivité, on préférera une constellation de grande taille composée d'agents délibératifs. De même si la limitation en termes de ressources est une contrainte forte, il est préférable d'utiliser des agents délibératifs. Cependant ces résultats se fondent sur le nombre de satellites mis en œuvre et non pas sur le nombre d'interactions au sein de la constellation. Une perspective d'étude serait alors de considérer la capacité d'extension d'un système non pas seulement en fonction du nombre d'agents qui le composent mais aussi en fonction de la quantité d'interactions entre les agents.

## Remerciements

Ce travail a été réalisé au sein du Département Commande des Systèmes et Dynamique du Vol de l'Onera en collaboration avec le Centre National d'Études Spatiales et Thales Alenia Space. Nous remercions Marie-Claire Charmeau (CNES) et Serge Rainjonneau (Thales Alenia Space) pour leur aide et leurs commentaires pertinents sur ce travail.

## 7. Bibliographie

- Bernon C., Gleizes M.-P., Peyruqueou S., Picard G., « ADELFE, a methodology for adaptive multi-agent systems engineering », *Engineering Societies in the Agents World*, vol. 3, p. 70-81, 2003.
- Bonnet G., Tessier C., « Collaboration among a Satellite Swarm », *Proceedings of the 6th International Conference on Autonomous Agents and Multi-Agents Systems*, p. 287-294, 2007a.
- Bonnet G., Tessier C., « Coopération au sein d'une constellation de satellites », *Actes des 15<sup>e</sup> Journées Francophones sur les Systèmes Multi-Agents*, p. 171-180, 2007b.
- Bonnet G., Tessier C., « Incremental adaptive organization for a satellite constellation », *Lecture Notes in Artificial Intelligence : Special Issue on Organized Adaptation in Multi-Agent Systems*, vol. 5358, p. 108-125, 2008.

- Brambilla A., Lavagna M., DaCosta A., Finzi A., « Distributed planning and scheduling for space system flotillas », *Proceedings of the 8th International Symposium on Artificial Intelligence, Robotics and Automation in Space*, 2005.
- Chow A.-H., « Parallel DEVS : a parallel hierarchical modular modeling formalism and its distributed simulator », *Transactions of the International Society for Computer Simulation*, vol. 13(2), p. 55-68, 1996.
- Contet J.-M., Gechter F., Gruer P., Koukam A., « Evaluation of global system state thanks to local phenomena », *Proceedings of the 18th European Conference on Artificial Intelligence*, p. 865-866, 2008.
- Damiani S., Verfaillie G., Charneau M.-C., « An Earth watching satellite constellation : how to manage a team of watching agents with limited communications », *Proceedings of the 4th International Joint Conference on Autonomous Agents and Multi-Agent Systems*, p. 455-462, 2005.
- Davidsson P., Johansson S., « Evaluating multi-agent system architectures : a case study concerning dynamic resource allocation », *Engineering Societies in the Agents World*, vol. 3, p. 25-77, 2003.
- Dutta P., Sen S., « Emergence of stable coalitions via task exchanges », *Proceeding of the 1st International Joint Conference of Autonomous Agents and Multi-Agent Systems*, p. 312-313, 2002.
- Giacomo P., « Sans dimension ou de dimension un ? », *Metrologia*, vol. 32, p. 311-313, 1996.
- Gupta I., Kermarrec A.-M., Ganesh A., « Efficient epidemic-style protocols for reliable and scalable multicast », *Proceedings of the 21st IEEE Symposium on Reliable Distributed Systems*, p. 180-189, 2002.
- Jenkins K., Hopkinson K., Birman K., « A gossip protocol for subgroup multicast », *Proceedings of the 21st International Conference on Distributed Computing Systems Workshops*, p. 25-30, 2001.
- Kubera Y., Mathieu P., Picault S., « La complexité dans la simulation multi-agent », *Actes des 15<sup>e</sup> Journées Francophones sur les Systèmes Multi-Agents*, p. 139-148, 2007.
- Lee L., Nwana H., Ndumu D., Wilde P. D., « The stability, scalability and performance of multi-agent systems », *BT Technology Journal*, vol. 16(3), p. 94-103, 1998.
- Parker L., « ALLIANCE : an architecture for fault-tolerant multi-robot cooperation », *IEEE Transactions on Robotics and Automation*, vol. 14(2), p. 220-240, 1998.
- Pittel B., « On spreading a rumor », *SIAM Journal of Applied Mathematics*, vol. 47, p. 213-223, 1987.
- Schetter T., Campbell M., Surka D., « Multiple agent-based autonomy for satellite constellation », *Artificial Intelligence*, vol. 145, p. 147-180, 2003.
- Shell D., Mataric M., « On foraging strategies for large-scale multi-robot systems », *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, p. 2717-2723, 2006.
- Sturm A., Shehory O., « A framework for evaluating agent-oriented methodologies », *Proceedings of the Workshop on Agent-Oriented Information System*, p. 60-66, 2003.
- Turner P., Jennings N., « Improving the scalability of multi-agent systems », *Proceedings of the Workshop on Infrastructure for Scalable Multi-Agent Systems*, 2000.

Walker J., « Satellite constellations », *Journal of British Interplanetary Society*, vol. 37, p. 559-571, 1984.

Zambonelli F., Jennings N., Wooldridge M., « Developing multiagent systems : the GAI A methodology », *ACM Transactions on Software Engineering and Methodology*, vol. 12(3), p. 317-370, 2003.

**ANNEXE POUR LE SERVICE FABRICATION**  
A FOURNIR PAR LES AUTEURS AVEC UN EXEMPLAIRE PAPIER  
DE LEUR ARTICLE ET LE COPYRIGHT SIGNE PAR COURRIER  
LE FICHER PDF CORRESPONDANT SERA ENVOYE PAR E-MAIL

1. ARTICLE POUR LA REVUE :

*RSTI - RIA – 23/2009. SMA auto-organisés et adaptatifs*

2. AUTEURS :

*Grégory Bonnet\* — Catherine Tessier\*\**

3. TITRE DE L'ARTICLE :

*Évaluation d'un système multirobot*

4. TITRE ABRÉGÉ POUR LE HAUT DE PAGE MOINS DE 40 SIGNES :

*Évaluation d'un système multisatellite*

5. DATE DE CETTE VERSION :

*25 août 2009*

6. COORDONNÉES DES AUTEURS :

– adresse postale :

\* Institut Charles Delaunay - Université de Technologie de Troyes

Équipe Environnement des Réseaux Autonomes

12, rue Marie Curie, BP 2060 - 10010 Troyes Cedex

gregory.bonnet@utt.fr

\*\* Onera - DCSD

Unité de Recherche Conduite et Décision

2, avenue Edouard Belin, BP 74025 - 31055 Toulouse Cedex 4

catherine.tessier@onera.fr

– téléphone : 05 62 25 29 14

– télécopie : 05 62 25 25 64

– e-mail : catherine.tessier@onera.fr

7. LOGICIEL UTILISÉ POUR LA PRÉPARATION DE CET ARTICLE :

L<sup>A</sup>T<sub>E</sub>X, avec le fichier de style `article-hermes.cls`,  
version 1.2 du 03/03/2005.

8. FORMULAIRE DE COPYRIGHT :

Retourner le formulaire de copyright signé par les auteurs, téléchargé sur :  
<http://www.revuesonline.com>

SERVICE ÉDITORIAL – HERMES-LAVOISIER  
14 rue de Provigny, F-94236 Cachan cedex  
Tél : 01-47-40-67-67  
E-mail : [revues@lavoisier.fr](mailto:revues@lavoisier.fr)  
Serveur web : <http://www.revuesonline.com>