



OATAO is an open access repository that collects the work of Toulouse researchers and makes it freely available over the web where possible

This is an author's version published in: <http://oatao.univ-toulouse.fr/18172>

To cite this version:

Bonnet, Jonathan and Gleizes, Marie-Pierre and Kaddoum, Elsy and Rainjonneau, Serge Planification de missions multi-satellites par système multi-agent coopératif. In: JFSMA'15 (23es Journées Francophones sur les Systèmes Multi-Agents), 29 June 2015 - 1 July 2015 (France).

Any correspondence concerning this service should be sent to the repository administrator: tech-oatao@listes-diff.inp-toulouse.fr

Planification de missions multi-satellites par système multi-agent coopératif

J. Bonnet^{a,b}

jonathan.bonnet@irt-saintexupery.com

M-P. Gleizes^b

Marie-Pierre.Gleizes@irit.fr

E. Kaddoum^b

Elsy.Kaddoum@irit.fr

S. Rainjonneau^a

serge.rainjonneau@irt-saintexupery.com

^aInstitut de Recherche Technologique Saint Exupéry, Toulouse, France

^bIRIT, Université Paul Sabatier - Toulouse III, France

Résumé

La planification de mission de constellations de satellites est un problème complexe soulevant d'importants défis technologiques pour les systèmes spatiaux de demain. L'augmentation de la taille des constellations, les performances évoluées et l'hétérogénéité des satellites sont tous des critères impliquant une combinatoire très élevée. Les techniques actuelles présentent des limites, car elles planifient un satellite à la fois et non la constellation dans son ensemble.

Dans cet article, nous proposons de résoudre ce problème difficile par les systèmes multi-agents coopératifs. Une amélioration du modèle AMAS4Opt est présentée, permettant de maximiser la coopération entre les agents du système. De par leurs interactions locales, ces agents, permettent d'obtenir une solution de bonne qualité en un temps raisonnable, en assurant un partage équitable des tâches au sein de la constellation. Enfin, une comparaison avec l'algorithme Glouton Chronologique, couramment utilisé dans le domaine spatial, souligne les avantages de l'approche proposée.

Mots-clés : Planification, Multi-satellites, Systèmes multi-agents coopératifs.

Abstract

Mission planning of constellation of Earth observation satellites is a complex problem raising significant technological challenges for tomorrow's space systems. The increasing size of constellations, the advanced performances and heterogeneity of satellites are different criteria involving a huge combinatorial search space. The techniques used today have limitations : they are planning one satellite at a time and not the whole constellation.

In this paper, we propose to solve this difficult problem by cooperative multi-agent systems. An improvement of the AMAS4Opt agent model is

presented to maximize the cooperation level. In the proposed system, the agents, through their local interactions, allow to reach a good solution in a reasonable time, while ensuring a fair distribution of tasks within the constellation. Finally, a comparison with the chronological greedy algorithm, commonly used in the spatial domain, underlines the advantages of the presented system.

Keywords: Scheduling, Multi-Satellite, Cooperative Multi-Agent Systems.

1 Introduction

Une constellation de satellites d'observation de la Terre est un ensemble potentiellement hétérogène de satellites. Elle permet de couvrir une large surface terrestre avec une fréquence de revisite élevée de chaque zone, tout en proposant des images de types différents et en garantissant la robustesse du système. Planifier une mission d'observation pour une telle constellation est une tâche difficile. En effet, beaucoup de paramètres et de contraintes souvent contradictoires sont à prendre en compte : le nombre de satellites et leurs caractéristiques, le volume des demandes des clients et les nombreuses contraintes qui y sont associées (le type de prise de vue, la priorité du client, la qualité demandée, la plage temporelle de validité, etc.) ainsi que les contraintes extérieures (la couverture nuageuse dans le cas de satellites optiques, etc.). La durée de mise en place d'un plan doit être raisonnable, notamment dans le cadre de demandes urgentes, typiquement inférieures à cinq minutes dans un contexte opérationnel, ce qui n'est pas toujours le cas actuellement.

Nous supposons connu l'algorithme qui permet à un satellite de construire son plan de mission

à partir d'un ensemble de prises de vues qui lui est affecté. Dans cet article nous nous intéressons au problème de la répartition des requêtes dans une constellation de satellites. Cette répartition consiste à planifier les demandes des clients sur les différents satellites, tout en respectant les contraintes et en assurant une charge équilibrée pour chaque satellite.

Pour cela, les systèmes multi-agents coopératifs (Gleizes, 2012) et leur capacité à prendre en compte un grand nombre d'entités et de contraintes sont une bonne approche pour résoudre ce type de problème. Dans ce travail nous nous appuyons sur AMAS4Opt (*Adaptive Multi-Agent System For Optimization*) (Kaddoum, 2011), un modèle d'agent générique qui fournit des patrons de conception pour résoudre des problèmes d'optimisation combinatoire à l'aide des systèmes multi-agents coopératifs.

La partie 2 de cet article formalise le problème et présente son contexte. La partie 3 développe le fonctionnement du système multi-agent ATLAS (*Adaptive saTellites pLanning for dynAmic earth obServation*). Enfin, la partie 4 présente une évaluation du système ATLAS ainsi que sa comparaison à un algorithme Glouton Chronologique.

2 Le problème de planification multi-satellite

Cette partie décrit le problème puis présente un résumé des méthodes de résolution actuellement utilisées.

2.1 La formalisation du problème

En nous appuyant sur les travaux de (Bensana and Verfaillie, 1999) et (Bonnet, 2008), nous allons tout d'abord présenter formellement notre problème. Dans la partie 3, nous nous appuyons sur cette modélisation pour *agentifier* le système.

Constellation de satellites. Une constellation est un ensemble potentiellement hétérogène de satellites, $Sat = \{s_1, s_2, \dots, s_n\}$, dans lequel chaque satellite s_i a ses propres caractéristiques :

- une trajectoire légèrement elliptique autour de la terre,
- une gestion d'énergie,
- une capacité mémoire,

- une charge utile, ici des instruments d'observation optique ou radar, et leurs attributs propres,
- un système de contrôle orbital et de gestion d'attitude, qui permet au satellite de contrôler son pointage (*vers la zone à observer*).

Requêtes. Une requête est une commande client. Nous noterons l'ensemble des requêtes à planifier $R = \{r_1, r_2, \dots, r_m\}$, avec r_i une requête définie par un ensemble de données :

- son type,
- une date de soumission,
- un intervalle de temps qui correspond à la période durant laquelle la requête est valable (*quelques heures à quelques jours, voire plusieurs semaines*),
- une zone géographique,
- une priorité donnée par le client,
- w , le taux de couverture nuageuse toléré, $0 \leq w \leq 1$,
- un ensemble de mailles, $M^i = \{m_1^i, m_2^i, \dots, m_p^i\}$, associé à la requête r_i .

Maille. Chaque requête est découpée en un ensemble de mailles dites *sœurs*, M^i . Une maille est une entité élémentaire que le satellite pourra acquérir en une seule prise de vue. Dans cet article, nous ne parlerons pas de l'algorithme de découpage des requêtes. Chaque maille reprend les contraintes de sa requête, mais est définie par une zone géographique plus petite. Une requête r_i est satisfaite si toutes ses mailles sont acquises. A chaque maille, est associé un ou plusieurs accès de visibilité correspondant à une période durant laquelle la maille est visible par le satellite. C'est dans cet accès de visibilité que l'acquisition sera effectuée. La durée du créneau d'acquisition est très inférieure à la durée de l'accès de visibilité. La figure 1 représente un accès de visibilité A sur la maille $M1$, par un satellite. Un créneau de réalisation C placé à l'intérieur de l'accès A . Le calcul de ces accès de visibilité est une donnée externe au problème.

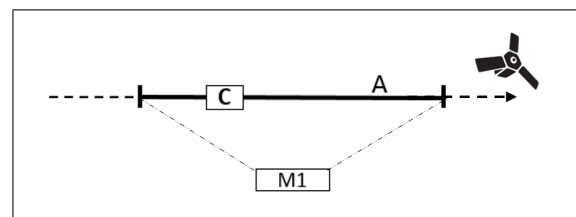


FIG. 1 – Représentation d'un accès de visibilité et d'un créneau de réalisation

Contraintes dures et souples. Deux catégories de contraintes sont à prendre en compte : les contraintes dures (la plage de validité ou le type d'image souhaité par le client par exemple) et souples (le taux de couverture nuageuse). L'ensemble C_h de contraintes dures doit être satisfait, les contraintes souples, C_s peuvent être relâchées.

But. L'objectif du problème est donc de trouver pour chaque satellite un ensemble de mailles, avec pour chaque maille une date d de couverture. Les mailles d'une même requête ne seront pas obligatoirement attribuées au même satellite. Les satellites devront assurer la couverture tout en :

- satisfaisant l'ensemble des contraintes dures C_h ,
- maximisant le nombre de contraintes souples C_s satisfaites,
- maximisant le nombre de mailles à satisfaire.

2.2 État de l'art

Différentes publications traitent du problème de la planification de mission d'une constellation, en le comparant à des problèmes d'optimisation tel que le problème du **sac à dos multidimensionnel** (Lemaître et al., 2002) et (Grasset-Bourdel et al., 2011) ou au problème du **voyageur de commerce** (Mancel, 2004). Le sac à dos représentant la capacité mémoire des satellites, et le chemin à trouver entre les villes modélisant l'enchaînement entre les acquisitions. De nombreux algorithmes existent pour résoudre ces problèmes, et les méthodes de planification spatiale s'en inspirent. Nous regroupons ici les approches les plus couramment utilisées dans le domaine, en commençant par les méthodes exactes puis les méthodes approchées.

Les méthodes exactes garantissent de trouver la solution optimale, si elle existe. La **Programmation Dynamique** est utilisée pour décomposer le problème en sous-problèmes plus simples. La difficulté de tels algorithmes vient de la possibilité de découper récursivement le problème initial. Comme le montre (Grasset-Bourdel, 2011), si on applique l'algorithme linéairement, son déroulement va rapidement utiliser la majorité des ressources mémoires et le rendre inapplicable en cas de requêtes stéréoscopiques (requêtes qui demandent des prises de vue d'un même endroit, mais avec un décalage temporel). (Mansour and Dessouky, 2010) émettent la même critique concernant des algorithmes construits sur une recherche de type

Séparation et Évaluation. En effet, même si toutes les solutions ne sont pas explorées car l'algorithme se sert des propriétés du problème pour guider la recherche, le déroulement d'une telle méthode est extrêmement coûteux (en espace mémoire et en temps de calcul). Ces méthodes ne sont donc pas adaptées pour des problèmes ayant de très grands espaces de recherche. Le temps de parcours et l'espace mémoire nécessaires étant dépendants de la taille de cet espace.

Les méthodes approchées peuvent être appliquées pour pallier ces différents inconvénients. Elles permettent de trouver une solution d'une bonne qualité en un temps raisonnable. La différence entre ces méthodes vient de l'heuristique qui guide la recherche et impacte donc la qualité de la solution finale.

L'algorithme le plus utilisé, car rapide et facile à mettre en oeuvre, est l'algorithme **Glouton Chronologique** (souvent nommée par son terme anglophone *Greedy Algorithm*). Son principe est simple, on planifie chronologiquement les requêtes, et en cas de conflit on applique une heuristique locale. De plus, cet algorithme sert souvent de base de travail, comme par exemple pour (Lemaître et al., 2002). Enfin, différentes méthodes approchées sont adaptées pour être appliquées au problème de planification de mission, comme les méthodes de **Recuit Simulé** (Wu et al., 2014), les **Recherches Tabou** (Bianchessi et al., 2007), ou bien les **Algorithmes Génétiques** tels que ceux présentés par (Globus et al., 2003) et (Mansour and Dessouky, 2010).

Tous ces algorithmes, même s'ils produisent des résultats plutôt satisfaisants présentent des limites. Tout d'abord, comme nous l'avons évoqué, ils dépendent fortement de l'heuristique qui guide la recherche. Il faut donc que le concepteur étudie et comprenne le problème dans sa globalité, et si celui-ci change, il faudra réadapter l'heuristique et l'algorithme. De plus, en cas d'ajout de requêtes pendant le déroulement du processus de planification, il faut reprendre le processus depuis la première opportunité de placement de la requête ajoutée. Enfin, ces algorithmes sont essentiellement conçus pour planifier la mission d'un satellite. Même si le problème reste globalement le même, ils font abstraction des nouvelles contraintes, comme par exemple l'équilibre de la charge de la constellation.

(Bonnet and Tessier, 2009) proposent une approche basée sur les systèmes multi-agents

pour la planification à bord des satellites d'une constellation. Dans cette approche, les satellites sont définis comme étant des agents communicants via des liaisons inter-satellites. L'objectif des satellites est de négocier entre eux la répartition des demandes de prises de vues reçues du sol permettant de maximiser le nombre de demandes satisfaites. Par l'utilisation des liaisons inter-satellites, l'applicabilité de cette approche requiert un certain nombre de conditions (taille de la constellation, distance entre satellites, etc.). Dans le travail présenté, nous proposons l'utilisation des systèmes multi-agents pour **la planification au sol de la mission d'une constellation de satellites**.

3 Le système ATLAS

Le modèle AMAS4Opt (Kaddoum, 2011) est basé sur l'approche par AMAS (*Adaptive Multi-Agent System*) (Gleizes, 2012). Dans cette approche, la coopération locale des agents permet au système de s'auto-organiser pour réaliser la fonction pour laquelle il est conçu. Ainsi, AMAS4Opt fournit les patrons de conception de deux rôles d'agents coopératifs : le rôle « contraint » et le rôle « service ». Les agents ayant le rôle « contraint » portent le problème et doivent être satisfaits, et les agents qui ont le rôle « service » possèdent les capacités et les compétences pour satisfaire les agents ayant le rôle « contraint ». Ce modèle utilise la *criticité* locale des agents ayant le rôle « contraint » comme moteur de la coopération entre agents. Nous avons utilisé ce modèle pour concevoir les agents et l'architecture générale du système et nous en proposons ici une amélioration.

Dans cette partie nous présentons les agents et leur rôle, puis la mesure de la criticité utilisée par les agents ayant le rôle « contraint ». Enfin, nous introduisons la mesure du coût représentant la criticité des agents ayant le rôle « service ».

3.1 Les agents

A l'aide de la modélisation du problème proposée dans la partie 2.1, et en se basant sur AMAS4Opt nous avons identifié neuf entités, parmi lesquelles :

- **trois types d'agents coopératifs** : le type agent requête, le type agent maille et le type agent satellite (leur comportement sera détaillé plus tard dans cet article),

- **trois types d'entités actives** : la couverture nuageuse, l'éphéméride solaire et les stations de télé-déchargement (ces entités n'ont pas de but à satisfaire),
- **trois types d'entités passives** (ressources du système) : la mémoire des satellites, leur batterie et un module de calcul de trajectoire et d'attitude orbitales.

L'utilisation du modèle AMAS4Opt permet de définir le comportement et les interactions des agents : les agents satellites prennent le rôle « service », et les agents requêtes et mailles le rôle « contraint ». Nous présentons ci-dessous les trois agents coopératifs en indiquant leur objectif local, les entités du système avec lesquelles ils seront en interaction au cours de leur processus de résolution et les agents avec lesquels ils seront amenés à négocier (échanges d'information, demandes de services, etc.), dans le but d'atteindre leur objectif.

Le type agent satellite (rôle « service »)

- objectif : acquérir des prises de vues,
- interactions avec :
 - le module de gestion trajectoire,
 - la couverture nuageuse,
 - la batterie,
 - la mémoire,
 - le soleil,
- négociations avec :
 - les agents mailles.

Le type agent requête (rôle « contraint »)

- objectif : avoir toutes ses mailles planifiées,
- négociations avec :
 - les agents mailles.

Le type agent maille (rôle « contraint »)

- objectif : être planifiée,
- négociations avec :
 - les agents satellites,
 - les agents requêtes.

3.2 La planification multi-satellite

Le système ATLAS assure la planification de la constellation grâce à la coopération de ses agents. Cette coopération est assurée via l'échange de messages entre les agents et est guidée par deux indicateurs : la criticité et le coût. C'est cette coopération qui permet de produire une planification respectant les critères suivant : un maximum de requêtes planifiées et un équilibre au sein de la constellation (les satellites doivent avoir tous la même charge de travail). Le fonctionnement du système ATLAS

TAB. 1 – Les agents coopératifs et leurs différents messages

Agent	Message	Signification
Satellite	estimationSlot(Cost c) confirmSlot(Booleen b)	Estimation du coût de la planification d'une maille Confirmation ou non de la planification
Maille	askForASlot(Access a) askConfirmationForASlot(Access a) informRequest()	Demande de planification Demande de confirmation pour l'accès Transmet de nouvelles informations à sa requête
Requête	informMesh()	Transmet de nouvelles informations à une maille

est basé sur la répétition par les agents du cycle « Perception - Décision - Action ». La phase de décision est l'étape centrale. En fonction de ses perceptions et de son état, l'agent choisit l'action à réaliser.

Les agents mailles envoient leurs demandes à tous les agents satellites ayant un accès de visibilité sur eux, et par la suite, privilégient l'agent satellite au coût le plus faible. Les agents satellites vont traiter tous les agents mailles les ayant sollicités en commençant par ceux les plus critiques. Les agents requêtes peuvent influencer la criticité de leurs agents mailles en fonction de l'avancement de la planification de la requête dans son ensemble. Les différents messages émis par les agents sont présentés dans la table 1.

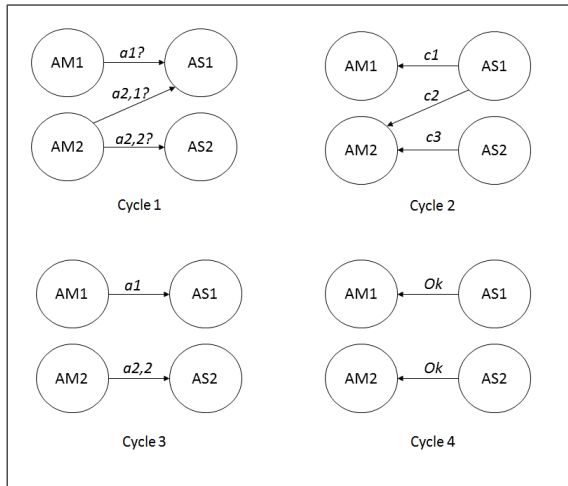


FIG. 2 – Exemple de planification

La figure 2 illustre le problème suivant : deux agents mailles (AM1 et AM2) communiquent avec deux agents satellites (AS1 et AS2). AM1 possède un seul accès de visibilité et AM2 deux. La résolution suit les cycles suivants :

- cycle 1 : AM1 et AM2 émettent pour tous leurs accès (a1, a2.1 et a2.2) des demandes aux agents satellites,
- cycle 2 : AS1 et AS2 classent les demandes

- en fonction de la criticité des agents mailles, et calculent les coûts (c1, c2 et c3) de planification qu'ils renvoient aux agents mailles,
- cycle 3 : AM1 et AM2 choisissent les agents satellites les moins coûteux et demandent confirmation,
- cycle 4 : AS1 et AS2 valident.

3.3 La criticité et le coût

La criticité. Dans ATLAS, le degré de non satisfaction des agents mailles est représenté par la criticité. (Bouziat et al., 2014) définissent la criticité comme « la distance qui sépare l'état courant [de l'agent], de l'état dans lequel son objectif local est atteint ».

Les règles de comportement local des agents satellites sont donc écrites pour favoriser les agents mailles les plus *critiques*. Cette criticité, qui indique donc le degré de non satisfaction de l'agent maille, est représentée par un ensemble de valeurs ordonnées. Dans la version actuelle du système, cette mesure comprend trois critères.

Le premier critère concerne l'urgence de la planification de la maille. Celle-ci est définie par le client au moment de sa commande. En cas d'égalité de ce critère, les agents satellites utilisent le nombre d'accès de visibilité restants (deuxième critère). Ainsi, un agent maille qui n'est visible qu'une seule fois par les satellites sera prioritaire par rapport aux autres agents mailles visibles plusieurs fois. En effet, si l'agent satellite privilégie un agent maille ayant plusieurs accès de visibilité sur un agent maille en ayant un seul, ce dernier ne pourra pas être planifié. Enfin, les agents mailles font évoluer leur criticité en fonction de leur état et des réponses à leurs messages envoyés aux agents satellites. Le troisième critère concerne donc l'influence des agents requêtes sur les agents mailles les composants.

L'évolution de la criticité d'un agent maille augmente si le nombre de ses créneaux de visibilité

diminuent, ou si ses mailles *sœurs* sont planifiées. Elle baisse lorsque l'agent maille est pris en compte. Finalement, elle est nulle lorsque l'acquisition de l'agent maille est planifiée.

Le coût. La criticité, telle qu'elle est définie dans le modèle AMAS4Opt, ne permet pas une coopération *totale* entre les agents : la coopération est seulement assurée par les agents ayant le rôle « contraint ». En effet, elle ne permet qu'aux agents ayant le rôle « service » de savoir quels agents ayant le rôle « contraint » sont prioritaires. Ainsi, un agent ayant le rôle « contraint » ne peut pas privilégier un agent ayant le rôle « service » sur un autre. Nous avons donc ajouté la notion de coût pour résoudre ce problème, et rendre les agents ayant le rôle « contraint » plus coopératifs.

Dans ATLAS, le coût est un indicateur sur la difficulté pour l'agent satellite à prendre en compte et planifier un agent maille. Il est calculé et retourné à l'agent maille ayant émis un message de demande de couverture. L'agent maille va donc recevoir un coût pour chaque demande émise aux agents satellites. Pour favoriser la coopération, un agent maille privilégie l'agent satellite lui répondant avec le coût le plus faible. Voici des éléments qui favorisent un coût élevé :

- une charge mémoire importante (beaucoup de mailles sont déjà planifiées par le satellite),
- un grand nombre de messages de demande de planification reçus,
- la nécessité d'adapter le plan, en déplaçant des éléments planifiés.

4 Résultats et discussions

Dans cette partie, nous présentons les résultats obtenus par le système multi-agent ATLAS pour planifier une constellation de satellites. Pour le tester nous avons développé un générateur de *scenarii*. Ces différents *scenarii* permettent de présenter l'intérêt du système ATLAS et de le comparer avec l'algorithme Glouton Chronologique couramment utilisé dans le domaine spatial.

4.1 Le générateur de solutions

Lors de l'établissement d'un plan de mission, les demandes des clients sont pré-traitées pour conduire à une liste de mailles correspondantes à acquérir. Cette phase de pré-traitement ne concerne pas ATLAS. Chacune de ces mailles terrestres est visible par un ou plusieurs satellites de la constellation. Chaque maille peut

donc être acquise par différents satellites à différents instants, ce qui augmente le nombre de solutions possibles.

Nous avons donc développé et utilisé un générateur pour construire des plans de mission *complets*, c'est-à-dire que tous les créneaux sont occupés par des acquisitions. Ces plans se concentrent sur les données temporelles des requêtes. Pour éviter de fournir une liste dans laquelle chaque requête n'a qu'un seul accès de visibilité, le générateur ajoute à chaque requête un nombre aléatoire d'accès de visibilité vers des satellites eux aussi tirés aléatoirement dans la constellation. Ainsi, nous obtenons une liste de requêtes ayant toutes plusieurs accès de visibilité. Chaque requête peut donc être couverte par N satellites et ceci à différents moments de la mission.

Ce générateur permet de générer un nombre conséquent de *scenarii* représentatifs de la combinatoire du problème réel. Cette génération a été validée par des experts du domaine spatial. Nous pouvons ainsi tester le système ATLAS, prouver sa validité, sa robustesse et son passage à l'échelle.

4.2 Expérimentations sur ATLAS

Nous allons tout d'abord nous intéresser à l'évolution de la criticité globale du système et des messages échangés. Pour cela, nous avons généré un scénario comprenant cinq satellites et cinq cents requêtes, scénario que nous nommerons S1. Les résultats que nous exposons sont obtenus en moyenne après avoir lancé cent fois S1.

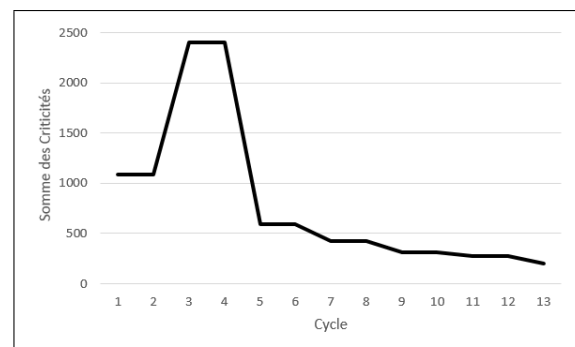


FIG. 3 – Évolution de la criticité globale d'ATLAS

Criticité du système. La figure 3 montre l'évolution de la criticité globale du système tout au long de son déroulement. Pour cela, nous avons donné une valeur numérique aux criticités des agents mailles. Cette valeur augmente

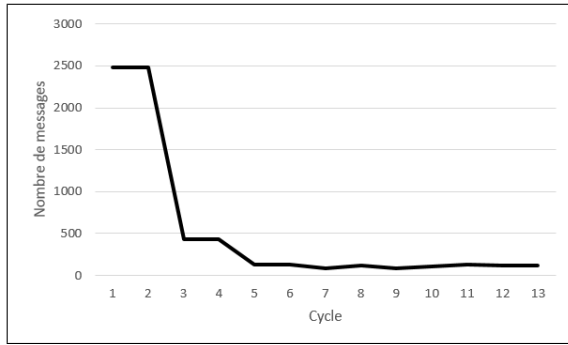


FIG. 4 – Évolution du nombre de messages par cycle

en fonction du nombre d'accès de visibilité restant et du nombre de messages envoyés, elle devient nulle quand la maille est planifiée. Après une croissance lors des premiers cycles, quand les agents mailles émettent des demandes de couverture aux agents satellites, la criticité globale ne fait que décroître avant de se stabiliser. La forte décroissance est due au grand nombre de planifications initiales. Le fait que la criticité globale décroît rapidement pour stabiliser vers une valeur assez faible montre qu'ATLAS converge rapidement vers une solution, où il reste peu d'agents mailles non planifiés.

Communications et cycles de résolution. La figure 4 permet de voir l'évolution du nombre de messages au cours des cycles de décision, et ce pour la résolution de S1. On note deux fortes décroissances entre les cycles 2 - 3 et 4 - 5. Les agents mailles initialisent le système, puis, ayant tous émis des requêtes vers les agents satellites au cycle 1, ce sont ces derniers qui traitent les demandes lors du cycle 2, les agents mailles étant en attente de réponses. Cinq fois moins de messages sont envoyés au cycle 3, en effet, à ce stade ce sont les agents mailles qui traitent les réponses à leurs requêtes et demandent confirmation au satellite qu'elles choisissent de favoriser. On note le même phénomène, un palier puis une forte diminution aux cycles 3, 4 et 5, mais avec moins de messages car davantage d'agents mailles sont déjà dans un état « planifiés », ils ne communiquent donc plus. En convergeant rapidement vers une solution, les agents n'encombrent pas le système avec un trop-plein de communication.

La table 2 permet de visualiser le nombre de messages émis par les agents mailles et le nombre de cycles de résolution nécessaires pour des constellations de plus en plus grandes. Un satellite est ajouté à chaque nouvelle constellation. La colonne S de la table 2 donne le

nombre de satellites pour chaque constellation. Les différents *scenarii* produits par le générateur durent 100 unités de temps, ce qui correspond à une moyenne de 86 requêtes pour chaque satellite, la croissance du nombre de requêtes est donc linéaire. Pour simplifier les expérimentations, nous avons décidé que chaque requête correspond à une maille. La proportion d'accès de visibilité pour chaque maille est donnée par un facteur en entrée du générateur, que nous avons augmenté en fonction du nombre de satellites, afin d'accroître la complexité du problème : rajouter des accès de visibilité revient à accroître le nombre de solutions possibles. L'augmentation de la complexité, et donc du nombre de messages émis, n'empêche pas le système de toujours converger vers une solution en un faible nombre de cycles et dans un temps de calcul restreint.

TAB. 2 – Nombre de messages échangés et nombre de cycles de résolutions

S	Mailles	Accès	Messages	Cycles
2	175	546	780	7
3	259	792	1 125	11
4	331	1 177	1 644	13
5	417	1 518	2 126	13
6	518	2 146	2 921	15
7	605	2 478	3 346	17
8	694	3 470	4 526	17
9	767	4 286	5 540	17
10	860	5 738	7 384	20

4.3 Comparaison avec un algorithme Glouton Chronologique : ChronoG

ChronoG. Pour analyser la qualité des solutions d'ATLAS, nous le comparons à la solution standard qui est actuellement utilisée par les références du spatial : l'algorithme Glouton Chronologique, nommé ici **ChronoG**. ChronoG traite la constellation un satellite à la fois. Nous supposons aussi qu'une maille sera acquise par un seul satellite.

L'algorithme suivant est donc répété pour chaque satellite : pour chaque pas de temps t de la durée de planification parcourue chronologiquement, ChronoG regarde si une maille est planifiée ou si l'espace est libre.

- Si une maille est planifiée, ChronoG passe à $t + 1$, sinon, ChronoG vérifie si une maille peut être placée à cet instant en déterminant si la maille en question possède un accès de visibilité qui englobe la date courante t .

- si plusieurs mailles sont possibles, ChronoG utilise une heuristique, détaillée ci-dessous, pour sélectionner une maille.
- en cas d'égalité, ChronoG choisira la maille ayant la plus petite différence entre la fin de son accès de visibilité et t .

Finalement, ChronoG planifie la maille choisie en réservant la durée nécessaire à son acquisition.

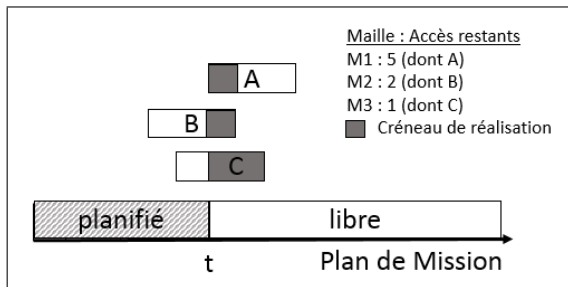


FIG. 5 – Heuristique de choix

La figure 5 permet de comprendre l'heuristique utilisée par ChronoG : les accès A, B et C (appartenant respectivement aux mailles M1, M2 et M3) sont possibles à t . C sera sélectionné car il appartient à la maille la plus critique : M3 n'a plus qu'un seul accès de disponible.

Expérience 1 : Taux de planification. La première expérimentation permet de comparer les systèmes ATLAS et ChronoG, sur des constellations de tailles différentes. Pour mener à bien cette expérimentation, nous avons généré aléatoirement neuf *scenarii*, en rajoutant à chaque fois à la constellation un satellite. De plus, nous augmentons progressivement le degré d'accessibilité des requêtes par les satellites, ainsi, plus la constellation est grande, plus les requêtes sont visibles. Cela permet de complexifier la résolution : en augmentant le nombre

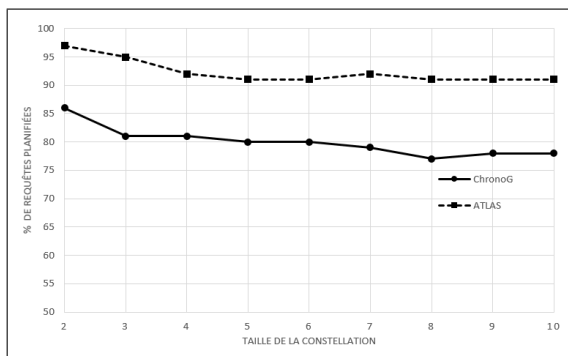


FIG. 6 – Comparaison des résultats entre ATLAS et ChronoG

d'accès possibles, le nombre de solutions possibles augmente aussi. Nous définissons un indicateur sur la solution : le pourcentage de requêtes planifiées. La solution proposée par le générateur est optimale sur ce critère, car toutes les requêtes sont planifiées : le taux de planification est de 100%. Ainsi il est facile de comparer facilement les deux systèmes, un taux élevé nous indiquant donc que la solution est de bonne qualité.

La figure 6 montre que les résultats obtenus avec le système ATLAS sont meilleurs. Certes, les deux premières exécutions (sur des constellations de deux et trois satellites et une moyenne de deux accès par maille) produisent de bons résultats sur les deux systèmes, mais cela s'explique par le fait que chaque maille possède peu d'accès de visibilité.

Quand la taille de la constellation et le facteur d'accessibilité augmentent, les résultats se stabilisent, ATLAS assurant un taux de planification avoisinant les 90%, alors que ChronoG est en moyenne à 80%. Cette stabilisation est plus visible sur les résultats produits par ATLAS où la convergence se fait dès que la constellation atteint 4 satellites. Enfin, il est important de noter que l'écart type des pourcentages de planification pour ATLAS est de l'ordre de 2%, et ce quelle que soit la taille de la constellation. ATLAS produit donc des solutions homogènes.

TAB. 3 – Comparaison taux de mailles planifiées et temps d'exécution

S	% planification		Temps d'exécution	
	ChronoG	ATLAS	ChronoG	ATLAS
2	86 %	97 %	20 ms	50 ms
3	81 %	95 %	31 ms	31 ms
4	81 %	92 %	32 ms	34 ms
5	80 %	91 %	32 ms	30 ms
6	80 %	91 %	47 ms	28 ms
7	79 %	92 %	50 ms	38 ms
8	77 %	91 %	64 ms	29 ms
9	78 %	91 %	50 ms	36 ms
10	78 %	91 %	50 ms	71 ms

La table 3 détaille les statistiques obtenues lors de l'exécution des neuf *scenarii* : taux de planification et temps moyen d'exécution. Ces résultats montrent qu'ATLAS planifie toujours plus de mailles que ChronoG. Concernant les temps moyens d'exécution, ATLAS est aussi meilleur. La variation de ces durées dépend de la complexité des *scenarii*. Un nombre de conflits important entraînent de nombreux appels à l'heu-

ristique ce qui ralentit ChronoG. A l'inverse, les agents du système ATLAS traitent localement ces conflits, ce qui ne ralentit pas la résolution.

Expérience 2 : Équilibre de charge. Pour illustrer cette seconde expérience, nous avons généré un scénario particulier. Dans ce dernier, deux satellites *agiles* identiques (que nous nommerons A et B) se suivent, cette agilité leur permet d'effectuer des prises de vues parallèles à leur déplacement, mais aussi en avant ou en arrière. De plus, ils possèdent les mêmes accès de visibilité. Ainsi, chaque maille peut être acquise par n'importe quel satellite. Le but ici est de montrer la distribution de la charge au sein de la constellation. Les figures 7 et 8 sont des représentations des tâches que les satellites devront effectuer durant leur passage. L'axe central représente la trace au sol, c'est-à-dire le déplacement vu du sol des deux satellites, avec sur la partie haute (rectangle noir) les tâches du premier satellite et sur la partie basse (damier) celles du second. Les rectangles sont hachurés en fonction du satellite responsable de l'acquisition. Les rectangles vides correspondent quant à eux, aux mailles non affectées.

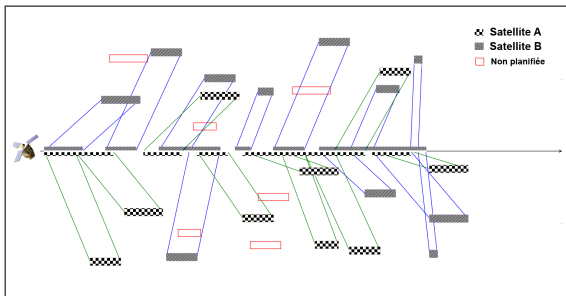


FIG. 7 – Mission planifiée avec ChronoG

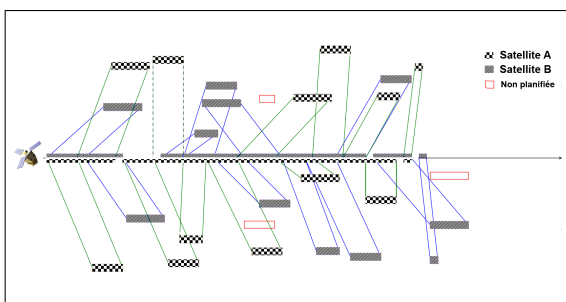


FIG. 8 – Mission planifiée avec ATLAS

Il est intéressant de noter que les résultats obtenus sur ce cas d'étude sont proches de ceux que l'on avait pu obtenir via des *scenarii* aléatoires, ATLAS fournissant toujours de meilleurs taux de planification (88% pour ATLAS contre 76% pour ChronoG). Pour ChronoG (figure 7),

le satellite A a seulement neuf tâches de planifiées et B en compte onze. En comparaison, nous voyons sur la figure 8 que la planification avec ATLAS donne douze prises de vues pour A et onze pour B. Enfin, nous constatons aussi que la planification via la méthode gloutonne entraîne davantage de périodes d'inactivités. En effet, ChronoG n'étant pas coopératif mais reposant sur une heuristique, les requêtes attribuées bloquent des enchaînements qui auraient permis d'ordonnancer davantage de tâches.

4.4 Discussion

Les systèmes actuels sont basés sur une heuristique globale et sont dépendants de cette dernière. Changer les entités du problème impose de modifier certains paramètres. Dans notre système multi-agent, ce sont la criticité des agents mailles et le coût de planification émit par les agents satellites qui guident la recherche. Or ces deux critères sont déduits du fonctionnement normal des entités et définis afin de mettre en avant la coopération, on peut donc changer les entités (les caractéristiques des satellites par exemple) sans avoir à les modifier.

Une autre limitation est l'approche *réductionniste* des méthodes utilisées pour planifier les constellations. Actuellement, les satellites sont planifiés individuellement, et ne tirent pas profit des avantages que présentent les constellations. Les agents composant ATLAS coopèrent pour planifier efficacement la constellation dans son ensemble. Le coût et la criticité sont donc là aussi deux puissants outils.

Enfin, l'ouverture et le dynamisme du système sont deux points intéressants. Dans les systèmes de planification actuellement utilisés, l'ajout et la suppression de requêtes durant le calcul du plan est une opération particulièrement difficile. C'est d'autant plus problématique dans une approche chronologique comme ChronoG : les requêtes sont traitées dans leur ordre d'accès. Supprimer ou ajouter une requête implique de recalculer le plan à partir de la date d'accès de la requête, et donc perdre du temps de calcul. Les requêtes urgentes, qui arrivent en cours de planification ne peuvent pas toujours être prise en compte, en fonction du temps restant. Au contraire, ATLAS est lui un système ouvert. L'ajout ou la suppression d'une requête entraîne une perturbation locale. Les agents vont la prendre en compte et s'adapter pour converger vers une nouvelle solution. Nos prochains travaux concerneront cette auto-adaptation.

5 Conclusion et perspectives

Dans cet article, nous avons présenté une approche basée sur les systèmes multi-agents adaptatifs pour planifier efficacement une constellation de satellites d'observation de la Terre : ATLAS. Dans ce système, la coopération des agents est guidée par la criticité des agents mailles, et par le coût indiqué par les agents satellites. Ces deux moteurs de la coopération permettent ainsi d'assurer qu'un nombre de requêtes important soit planifié de façon équilibrée entre les satellites de la constellation. Les différentes expérimentations ont montré qu'ATLAS fournissait des résultats de meilleure qualité que l'algorithme Glouton Chronologique couramment utilisé (ChronoG). De plus, ATLAS fournit des solutions ayant un meilleur équilibrage de la charge de la constellation. Ces premiers résultats sont donc très encourageants et prouvent l'intérêt d'une approche par système multi-agent pour planifier efficacement une constellation de satellites.

Nos futurs travaux vont concerner la prise en compte de la dynamique et l'auto-adaptation du système. Nous envisageons aussi de tester ATLAS sur des cas d'utilisations réels. Ainsi, nous pourrions comparer nos résultats avec des plans de mission produits par le CNES (*Centre National d'Études Spatiales*) ou Airbus D&S-Geo (anciennement Spot Image), deux organismes qui gèrent les constellations des satellites Spot et Pléiades.

Remerciements

Les auteurs souhaitent remercier l'IRT Saint Exupéry pour le financement de cette recherche.

Références

- Bensana, E. and Verfaillie, G. (1999). Earth Observation Satellite Management. In *Constraints*, volume 299, pages 293–299.
- Bianchessi, N., Cordeau, J. F., Desrosiers, J., Laporte, G., and Raymond, V. (2007). A heuristic for the multi-satellite, multi-orbit and multi-user management of Earth observation satellites. *European Journal of Operational Research*, 177(2) :750–762.
- Bonnet, G. (2008). *Coopération au sein d'une constellation de satellites*. PhD thesis.
- Bonnet, G. and Tessier, C. (2009). Évaluation d'un système multirobot cas d'une constellation de satellites. *Revue d'Intelligence Artificielle*, 23 :565–593.
- Bouziat, T., Combettes, S., Camps, V., and Glize, P. (2014). La criticité comme moteur de la coopération dans les systèmes multi-agents adaptatifs (short paper). In *Journées Franco-phones sur les Systèmes Multi-Agents, 2014*, pages 149–158.
- Gleizes, M.-P. (2012). Self-adaptive Complex Systems (regular paper). In *European Workshop on Multi-Agent Systems, Maastricht, The Netherlands*, volume 7541, pages 114–128.
- Globus, A., Crawford, J., Lohn, J., and Pryor, A. (2003). Scheduling earth observing satellites with evolutionary algorithms. In *Conference on Space Mission Challenges for Information Technology*.
- Grasset-Bourdel, R. (2011). Planification dynamique et réactive pour des satellites agiles d'observation de la Terre.
- Grasset-Bourdel, R., Flipo, A., and Verfaillie, G. (2011). Planning and replanning for a constellation of agile Earth observation satellites.
- Kaddoum, E. (2011). *Optimization under Constraints of Distributed Complex Problems using Cooperative Self-Organization*. PhD thesis.
- Lemaître, M., Verfaillie, G., Jouhaud, F., Lachiver, J. M., and Bataille, N. (2002). Selecting and scheduling observations of agile satellites. *Aerospace Science and Technology*, 6(5) :367–381.
- Mancel, C. (2004). *Modélisation et résolution de problèmes d'optimisation combinatoire issus d'application spatiales*. PhD thesis.
- Mansour, M. A. and Dessouky, M. M. (2010). A genetic algorithm approach for solving the daily photograph selection problem of the spot5 satellite. *Computers & Industrial Engineering*, 58(3) :509–520.
- Wu, G., Wang, H., Li, H., Pedrycz, W., Qiu, D., Ma, M., and Liu, J. (2014). An adaptive Simulated Annealing-based satellite observation scheduling method combined with a dynamic task clustering strategy. *Computing Research Repository*, abs/1401.6098 :23.