

On-board cooperation for satellite swarms

Grégory Bonnet and Catherine Tessier

ONERA / DCSD 2 av. Edouard Belin 31000 Toulouse FRANCE

{gbonnet,tessier}@onera.fr

Abstract

The paper deals with on-board planning for a satellite swarm via communication and negotiation. We aim at defining individual behaviours that result in a global behaviour that meets the mission specifications. We will present the formalization of the problem, a solving method based on reactive decision rules, and first results.

1 Introduction

Much research has been undertaken to increase satellite autonomy such as enabling them to solve problems that may occur during a mission, adapting their behaviour to new events and transferring planning on-board ; even if the development cost is increased, there is an increase in performance and mission possibilities [Polle, 2002]. Moreover, the use of satellite swarms - i.e. sets of satellites flying in formation or in constellation around the Earth - makes it possible to consider joint activities, to distribute skills and to ensure robustness. The objective of this work is to use intersatellite links (ISL) in an Earth observation constellation inspired from the Fuego mission [Damiani *et al.*, 2005], in order to increase the system reactivity and to improve the global return.

2 Problem characterization

The observation satellite constellation is composed of homogeneous satellites that are equipped with a single observation instrument and a detection instrument that allows to generate on-board requests. Given the satellite orbits, it is possible to consider that two (or more) satellites meet in the polar areas, and thus can communicate without the ground intervention. Intuitively, this intersatellite communication increases the reactivity of the constellation.

The features of the problem are the following:

- 3 to 20 satellites (the agents) in the constellation;
- pair communication around the poles;
- no ground intervention during the planning process;
- asynchronous requests coming from the ground or generated on-board with various priorities.

2.1 Satellite swarm

An observation satellite swarm is a multi-agent system where the requests do not have to be carried out in a fixed order and the agents do not have any physical interaction. Carrying out an observation cannot prevent another agent from carrying out another one, even the same one.

Definition 1. A swarm $\mathcal{E}$ is a triplet $\langle \mathcal{S}, \tau, Vicinity \rangle$:

- $\mathcal{S}$ is a set of n agents $\{s_1 \dots s_n\}$;
- $\tau \subset \mathbb{R}$ is a common clock;
- $Vicinity : \mathcal{S} \times \tau \mapsto 2^{\mathcal{S}}$.

For a given agent at a given time, the *vicinity relation* returns the set of agents with which it can communicate. This link exists when the agents meet.

2.2 Requests

A *request* is an observation that must be carried out.

Definition 2. An request R is a tuple $\langle id_R, r, b_R, S_R, t_R \rangle$:

- id_R is an identifier;
- r is the set of the characteristics¹ of R ;
- $b_R \in \{true, false\}$ specifies if R has been realized;
- $S_R \subseteq \mathcal{S}$ is the set of agents knowing R as it is;
- $t_R \in \tau$ is a timestamp.

Each pair (request, agent) is associated with a utility value - noted $u_R^{s_i}$ - representing the capability of the agent to realize the request at a date close to a desired date of observation, and its capability to quickly download it after its realization.

2.3 On board planning

Let us suppose that each agent is equipped an on-board planner. Given a set of requests and a utility value for each of them, an individual agent is supposed to be able to generate a plan. This plan is the set of requests that the agent expects to carry out. As we are in a cooperative context, these individual plans must be revised in order to take the others' plans into account. Inspired from contingency planning [Meuleau and Smith, 2003], the agents' plans are made up of unquestionable requests and uncertain ones on which a decision will be made at the end of the decision horizon, which is the sliding deadline for the request realization.

¹Such as the priority, position and desired date of observation.

2.4 Problem

Then, the problem is the following: we would like each agent to build request allocations dynamically such as if these requests are carried out, their number is the highest possible or the global utility is maximal. Let us notice that both criteria are not necessarily compatible.

3 Communication protocol

As the choices of an agent will be influenced by the choices of the others, the agents have to reason on common pieces of knowledge about the requests. Therefore an effective communication protocol has to be set up.

3.1 Requests and candidacies

Two kinds of knowledge must be shared : knowledge about requests and knowledge about the others' intentions, i.e the others' candidacies.

Definition 3. A candidacy C is a tuple $\langle s_C, R_C, M, S_C, t_C \rangle$:

- $s_C \in \mathcal{S}$ is the candidate agent;
- R_C is the request on which s_C candidates;
- M is the modality: s_C commits to realize R_C or intends to realize R_C or intends not to realize R_C ;
- S_C is the set of agents knowing R as it is;
- t_C is a temporal timestamp;

Candidacies are generated by the on-board planner. Consequently, each agent has two knowledge bases, a set of requests and a set of candidacies, that will be shared with the others' through the communication protocol.

3.2 An epidemic protocol

It is defined as an epidemic protocol [Holliday *et al.*, 2000] based on overhearing [Legras and Tessier, 2003]. The idea is to benefit from all the communication opportunities even if transmitted information does not concern the agents directly. An agent notifies any change within its knowledge bases and each agent propagates these changes to its vicinity who update their knowledge bases and reiterate the process. In order to know what information must be transmitted, each agent keeps the list of agents knowing each request (S_R) and candidacy (S_C). Updates are realized thanks to the temporal timestamps (t_R and t_C) affixed to each information.

4 Collaboration

4.1 Conflicts

When two agents compare their respective plans some conflicts may occur. It is a matter of redundancies between allocations on a given request, i.e.: several agents are candidates to carry out the request. Whereas this redundancy can sometimes be useful to ensure the realization of a request (because an observation may not be successful), it can also lead to a loss of opportunity. Consequently, conflict allows us to define uncertain requests and parts of the plan that can be revised.

Definition 4. Let C_1 and C_2 two candidacies. C_1 is in a conflict with C_2 iff $s_{C_1} \neq s_{C_2}$ and $R_{C_1} = R_{C_2}$.

4.2 Reactive strategies

When they are in a conflict, the agents must find a local agreement by using the conflict in order to increase the number of observations, or to increase their quality or to make sure that a request will be carried out. Because of the lack of time during encounters between agents and the computational power in space, reactive decision rules are considered. The following behaviours (strategies) are considered to reach an agreement:

- *Expertise*: agent with the highest utility keeps its candidacy.
- *Opportunism*: agent with the most resources keeps it.
- *Insurance*: both agents maintain their candidacies.

5 Experimentations and first results

A simulator including the communication protocol and collaboration strategies has been implemented in JAVA with the JADE platform [Bellifemine *et al.*, 1999]. The scenario is based on 3 satellites, 40 requests and a 6 hour-orbit simulation. Firstly, a witness simulation (without collaboration strategies) has been launch. Then simulations using strategies have been performed.

Simulation	Observations	Redundancies	Messages	Average utility
Witness	38	5	1079	190, 39
Strategies application	33	0	1079	202, 6

We can notice that using strategies reduces redundancies without noticeable consequences on the number of observations and on the average utility. Every avoided redundancy is a gain for future requests generated during the simulation.

6 Conclusion and further works

In a space context, communication and computation time are lacking for cooperation. However, simple decision rules can be used to solve this problem. We have presented a communication protocol and a coordination method dedicated to a satellite swarm. First experiments are promising: cooperation reduces redundancies and consequently useless activities. Further works consist in proposing new strategies to take into account new constraints (e.g. satellite failures).

References

- [Bellifemine *et al.*, 1999] F. Bellifemine, A. Poggi, and G. Rimassa. JADE - a FIPA-compliant agent framework. In *Proceedings of PAAM'99*, pages 97–108, 1999.
- [Damiani *et al.*, 2005] S. Damiani, G. Verfaillie, and M.-C. Charneau. An earth watching satellite constellation : How to manage a team of watching agents with limited communications. In *Proceedings of the 4th AAMAS*, pages 455–462, 2005.
- [Holliday *et al.*, 2000] J. Holliday, D. Agrawal, and A. ElAbbadi. Database replication using epidemic communication. In *Proceedings of the 6th Euro-Par Conference*, pages 427–434, 2000.
- [Legras and Tessier, 2003] F. Legras and C. Tessier. Lotto: group formation by overhearing in large teams. In *Proceedings of 2nd AAMAS*, 2003.
- [Meuleau and Smith, 2003] N. Meuleau and D.E. Smith. Optimal limited contingency planning. In *Proceedings of the 19th UCAI*, pages 417–426, 2003.
- [Polle, 2002] B. Polle. Autonomy requirement and technologies for future constellation. *Astrium Summary Report*, 2002.