# Planning for an Ocean Global Surveillance Mission

**C. Pralet** and **G. Verfaillie**
ONERA, Toulouse, France
FirstName.LastName@onera.fr

**X. Olive** and **S. Rainjonneau**
TAS, Toulouse, France
FirstName.LastName@thalesaleniaspace.com

**I. Sebbag**
CNES, Toulouse, France
FirstName.LastName@cnes.fr

### Abstract

In this paper, we present the problem of planning the activities of a constellation of radar satellites in order to fulfil as well as possible an ocean global surveillance mission. Then, we describe the two greedy algorithms that have been designed and implemented so far to solve the huge daily problem instances: a chronological algorithm and a non chronological one. We report first experimental results and conclude with what should be done beyond this preliminary work.

## Introduction

In this paper, we consider a new kind of Earth observation mission whose objective is the regular surveillance of ship movements over all the oceans: all the ocean areas must be covered at least once every day, more often if possible, using a wide swath observation mode; moreover, some special areas, known only every day before planning, must be compulsorily observed at specified times, using a narrow swath mode.

To perform such a mission, it is planned to use a constellation of four satellites. These satellites would be placed on different orbital planes and on circular, low altitude, quasi-polar orbits. They would be all equipped with a radar observation instrument usable in different modes, a mass memory, and a high rate antenna, allowing them to observe, to record data, and to download this data to ground stations.

Because the planning problem to be solved every day is huge, we first designed and implemented two greedy algorithms using two different plan construction principles: a chronological one and a non chronological one.

Experiments carried out on a realistic scenario allowed us to conclude that at least greedy algorithms can be practically used to build every day executable constellation activity plans of reasonable quality. This paves the way for the design of more sophisticated local search algorithms able to improve plan quality.

In this paper, we first describe the mission and the physical system that is provided for fulfilling it. Then, we define and analyze the daily planning problem to be solved. We describe the two planning algorithms that have been designed and implemented so far. After reporting experimental results, we conclude with directions for future work.

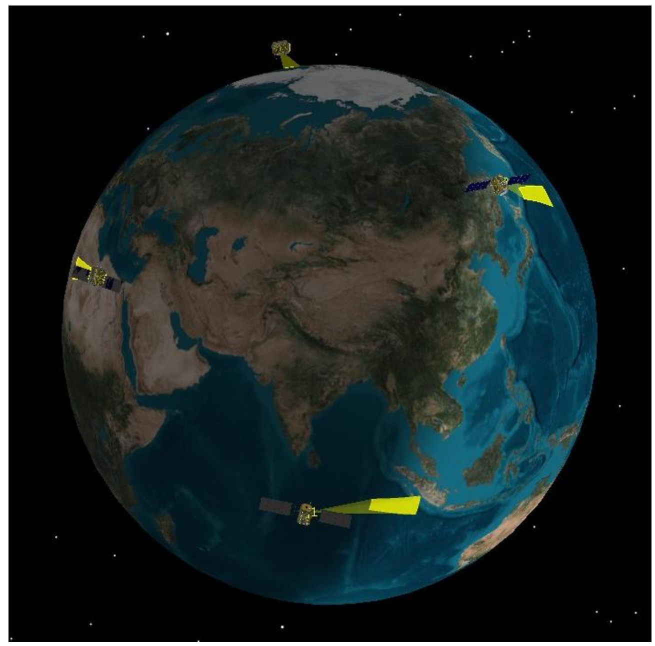

Figure 1: Four satellite constellation.

## Mission

The context of this work is the SAMSON project, currently under study at CNES (French Space Agency), whose objective is the design and development of a space system able to look regularly after ship movements over all the oceans.

In fact, the mission objective is twofold: first, to visit every ocean area at least once every day, more often if possible, using a wide swath observation mode (surveillance mission); second, to observe a small number of areas of special interest at specified times, using a narrow swath mode (observation mission).

To manage the regular surveillance mission, all the oceans are split into small 100km x 100km rectangle areas, resulting in about 37,000 areas to be visited every day. Because areas do not have all the same importance (for example, areas covering strategic ship roads may be more important than others), a weight is associated with each of them.

Areas of special interest are only known every day before planning for the next day. Their observation at specified times is compulsory.

## Physical system

**Constellation** To fulfil such a mission, it is planned to use a constellation of four satellites, placed on different orbital planes and on circular, low altitude, quasi-polar orbits. Orbiting around Earth takes about one hour and a half.

**Instrument modes** Each satellite is equipped with a radar observation instrument usable in four different modes: a wide swath regular surveillance mode (SURV), two narrow swath high resolution observation modes (HR1 and HR2), and a stand-by mode (SB). We will consider that the instrument is ON when it is in SURV, HR1, or HR2 mode and is OFF when it is in SB mode. Mode transitions are instantaneous.

**Surveillance** When the instrument is in mode SURV, it scans a wide strip (about 1000km wide) on Earth surface, on the right of the satellite track, which covers concurrently a number of the areas that result from ocean splitting.

Figure 1 is an artist view of the four satellites with, for each of them, the scanning (in yellow) performed by the radar instrument on the right of the satellite track over a very short time period. Figure 2 shows the scanning performed by the radar instruments of the four satellites over a 45 minute period (a color is associated with each satellite of the constellation). Figure 3 focuses on one satellite and on the ocean splitting (each point on the ocean surface represents the center of an area). Finally, Figure 4 is a schematic view of the ocean splitting, of the satellite track, of the radar instrument swath, of an area, and of its associated area visit window.

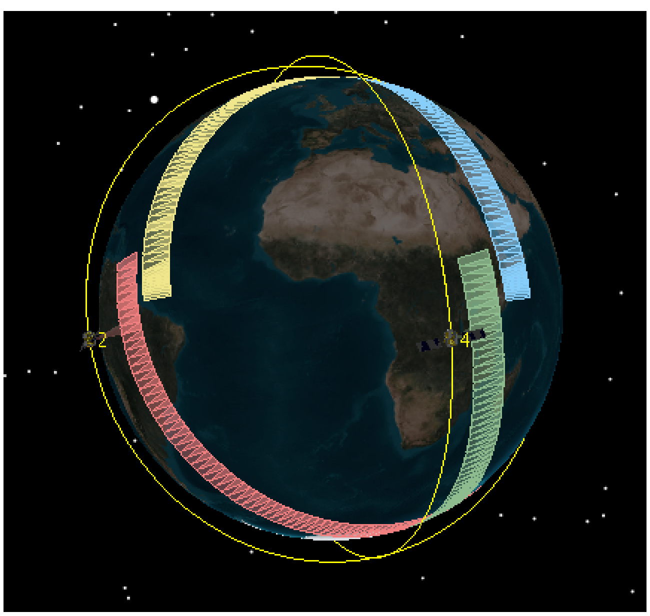

Figure 2: Scanning performed by the radar instruments of the four satellites over a 45 minute period.

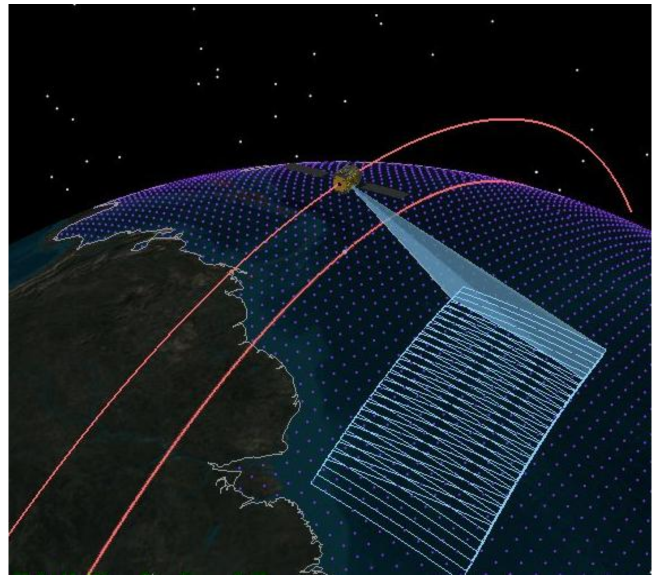

Figure 3: Ocean splitting and scanning performed by the radar instrument of one satellite over a short time period.

**Observation and attitude movements** In order to increase the number of observation opportunities in mode HR1 or HR2, each satellite would be able to perform attitude movements in order to observe on the left of the satellite track as well as on the right. No observation and no surveillance is possible during a satellite attitude movement, but the instrument can remain ON (in mode SURV), in order to limit the number of ON/OFFs. Between two successive observations on the left, surveillance on the left is possible too, in order to avoid two attitude movements (one to go from the left to the right and one to come back to the left). However, by default, satellites perform observation and surveillance on the right.

**Energy and memory** Each instrument mode is characterized by an instantaneous consumption of energy and memory. On-board energy is produced by solar panels when the satellite is not in eclipse until a maximum level that depends on battery capacity. It must never be below a minimum level. On-board available mass memory is released by data downloads when the satellite is within a visibility window of a ground reception station. It must never be below 0 (memory overwriting). To be sure to meet these constraints, a maximum consumption of energy and memory per satellite orbit is enforced.

**Instrument temperature** Instrument temperature evolution depends on the fact that the instrument is ON or OFF and on the fact that the satellite is in eclipse or not. It cannot be below a minimum level and must never be above a maximum level.

**Number of ON/OFFs** For the sake of long-term reliability, the number of instrument ON/OFFs over the whole planning horizon (typically one day) is limited.

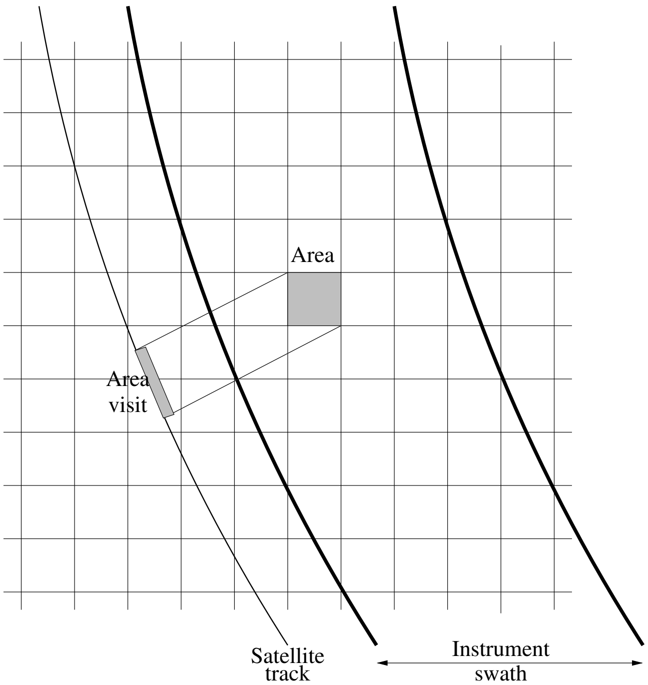


Figure 4: Ocean splitting, satellite track, radar instrument swath, area, and its associated area visit window.

## Management system

The constellation of satellites would be managed from the ground as follows: the requirements in terms of surveillance remain the same day after day; only the requirements in terms of observation change; each day, before planning for the next day, the compulsory observations are known; each one is characterized by a satellite, a starting time, a duration, a side (either right or left), and a mode (either HR1 or HR2); they are assumed to be compatible (performing all of them is possible); the problem is to organize as well as possible the surveillance mission while satisfying all the physical constraints and guaranteeing the execution of all the compulsory observations; when a executable plan has been built on the ground in the mission center, it is uploaded to all the satellites for execution.

## Planning problem

For the moment, we only consider observation and surveillance activities and do not consider data downloads.

**Data** Planning problem data is the following one:

- the static parameters of the physical system (for example, the duration of a satellite attitude movement);
- the planning horizon (for example, the next day);
- for each satellite:
  - its initial state (at the beginning of the planning horizon), including its eclipse status, its observation side, and its instrument mode and temperature;
  - the times of orbit change and of eclipse status change (over the planning horizon);
  - the set of compulsory observations with, for each of them, its starting time, its duration, its side, and its mode;
  - the set of surveillance opportunities with, for each of them, its area, its starting time, its duration, and its side.

**Variables** In classical Earth observation planning problems (see for example (Verfaillie and Lemaître 2006) for a tutorial and an associated commented bibliography), observations are mutually exclusive: two observations cannot be performed concurrently. In such conditions, it is relevant to associate with each candidate observation $o$ decision variables that represent the fact that $o$ is realized or not and, in case of realization, $o$'s realization parameters (time, mode ...).

On the contrary, in the planning problem we face, surveillance of different areas can be performed concurrently. In such conditions, it is more relevant to consider decision variables that represent satellite activity (time, mode, side ...).

We define an ON interval as an interval on which the instrument is ON (in mode HR1, HR2, or SURV) and an OBS interval as an interval on which observation or surveillance is effective (no attitude movement).

In such conditions, the decision variables we consider are the following ones for each satellite:

- the number of ON intervals (over the planning horizon);
- for each ON interval:
  - its starting time;
  - its duration;
  - the number of OBS intervals it includes;
  - for each OBS interval:
    - ∗ its starting time;
    - ∗ its duration;
    - ∗ its side.

Figure 5 shows an example of plan including 3 ON intervals and 6 OBS ones.

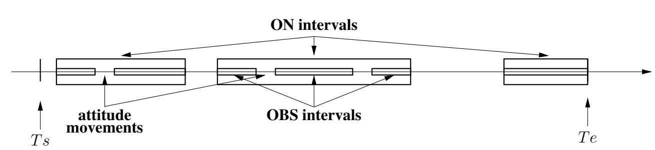


Figure 5: Example of plan including 3 ON intervals and 6 OBS ones over the planning horizon $[Ts, Te]$.

**Constraints** The set of constraints to be considered derives from this choice of variables and from the requirement that all the compulsory observations be performed and all the physical constraints described in Section *Physical system* be satisfied: temporal constraints on ON and OBS intervals, energy, memory, and temperature constraints, and limitations on the number of ON/OFFs.

**Optimization criterion** We propose to consider the following optimization criterion:

$$c = \sum_{a \in A} (w_a \cdot (1 - \alpha^{nv_a}))$$

where $A$ is the set of areas to be visited, $\alpha$ is a parameter to be set strictly between 0 and 1 and, for each area $a \in A$, $w_a$ is its weight and $nv_a$ the number of times it is visited over the planning horizon (number which can be computed from the set of OBS intervals in the plan).

As any optimization criterion, such a criterion is arguable. It is however justified by the following properties: for each area $a \in A$, let be $c_a = w_a \cdot (1 - \alpha^{nv_a})$; if area $a$ is not visited ($nv_a = 0$), then $c_a = 0$; if it is visited once ($nv_a = 1$), then $c_a = w_a \cdot (1 - \alpha)$; if it is visited twice ($nv_a = 2$), then $c_a = w_a \cdot (1 - \alpha^2)$; if $nv_a \to +\infty$, then $c_a \to w_a$; each new visit of area $a$ returns $w_a \cdot (1 - \alpha) \cdot \alpha^{nv_a}$; moreover, local $c_a$ values are added over all the areas to be visited.

**Instance size** The instances associated with a planning problem over an one day horizon typically involve about 30 compulsory observations, about 37,000 areas to be visited, and on average 4.93 surveillance opportunities per area.

## Possible approaches

**Constraint programming** Because the set of relevant starting and ending times for ON and OBS intervals is finite (set of starting and ending times for compulsory observations and surveillance opportunities), the planning problem we face can be modeled as a constrained optimization problem. Hence, classical constraint programming can be used to model and solve it.

This is the first approach we followed, using the OPL tool (*Optimization Programming Language*, now referred to as *IBM ILOG CPLEX Optimization Studio* (IBM ILOG )). This allowed us to get quickly a reference executable model. However only very small artificial instances, involving some tens of areas and some surveillance opportunities per area, could be effectively optimally solved using such a generic optimization tool.

**Dynamic programming** The planning problem we face can be also been modeled as a sequential decision-making problem, with a finite horizon and deterministic transitions (no uncertainty), which can be solved using dynamic programming algorithms (Bellman 1957).

However, such an approach would require to consider the following state variables: for each satellite, current time, mode, side, energy, memory, temperature, and remaining number of ON/OFFs and, for each ocean area, number of visits already performed. As a consequence, mainly because of the number of areas, the state space to be considered would be too huge and the approach impractical.

Such a situation would change and dynamic programming could become practical if the optimization criterion would be additive, for example if $c = \sum_{a \in A} (w_a \cdot nv_a)$.

## Two planning algorithms

To solve at least approximately real-size instances we developed two specific greedy algorithms, using two different plan construction principles: a chronological one and a non chronological one.

### A chronological greedy algorithm

The first algorithm builds a plan in a chronological way by going forward from the beginning to the end of the planning horizon.

At any step of the algorithm, the system is in some state defined by (i) the current satellite, (ii) for each satellite, by the current decision time, mode, side, energy, memory, temperature, and remaining number of ON/OFFs and, (iii) for each ocean area, by the number of visits already performed. At the beginning of the algorithm, the system is in the state it reached at the end of the previous planning horizon.

At any step, when the current satellite is $s$ and the current time is $t$, a first decision must be made concerning $s$ on the next area visit $v$ to be performed. To make this decision, the next compulsory observation $o$ to be performed by $s$ is computed, as well as the set $V$ of area visits that can be performed by $s$ between $t$ and the beginning of $o$ (ordered by increasing ending times). If $V$ is not empty, the first area visit $v$ in $V$ is candidate and one has to decide on performing it or not. If $V$ is empty, $o$ must be performed (no other choice). See Figure 6 for an illustration.

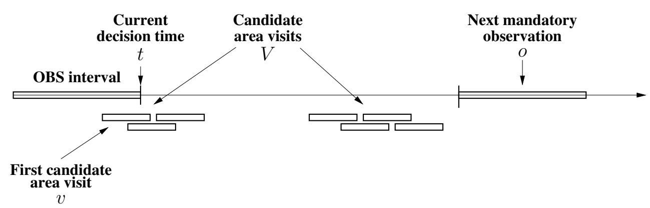


Figure 6: Current decision time $t$, next compulsory observation $o$ to be performed, and set $V$ of candidate area visits before $o$.

If it has been decided to perform $v$, a second decision (in fact, a sequence of decisions) must be made on what will be done before and just after performing $v$ in terms of instrument activation and attitude movement: to set or to maintain the instrument ON between $t$ and the beginning $t'$ of $v$; if $v$ is a compulsory observation to be performed on the left and the next compulsory observation must be performed on the left too, to remain on the left or to come back to the right at the end of $v$; in the latter case, to maintain or not the instrument ON during the attitude movement. See Figures 7 and 8 for an illustration.

Once all these decisions have been made, the number of visits of the area $a$ that is associated with $v$ (and possibly of other areas if it is decided to have the instrument ON between $t$ and $t'$) can be incremented; the end $t''$ of $v$ (taking into account the duration of the possible attitude movement at the end of $v$) becomes the next decision time for satellite

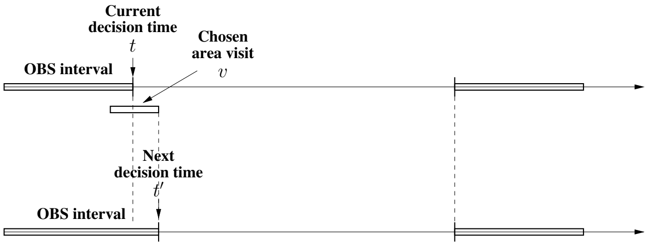


Figure 7: Chosen area visit $v$ and next decision time $t'$.

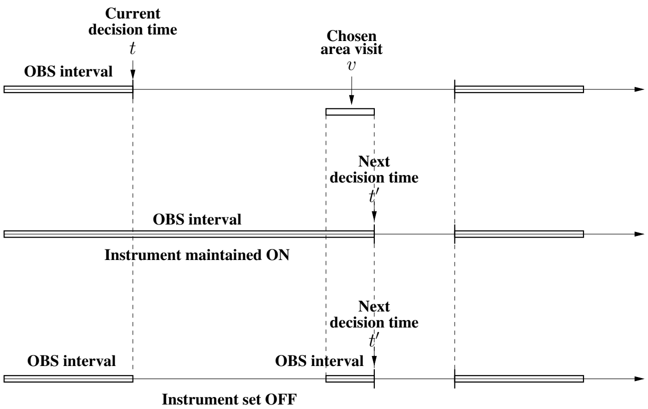


Figure 8: Chosen area visit $v$ and next decision time $t'$, with two options: to maintain the instrument ON or to set it OFF between $t$ and $t'$.

$s$; the state of $s$ at $t''$ can be computed; if this state is inconsistent (too much energy consumption, too high instrument temperature, impossibility of performing all the remaining compulsory observations ...), decisions are cancelled and other ones are made; in case of consistency, the next algorithm decision step is defined by the earliest decision time over all the satellites.

This algorithm contains some choice points at which local heuristics can be used in order to guide the search towards plans of reasonable quality. The following heuristics have been implemented and used in experiments:

- for the choice of performing or not a candidate area visit $v$, let $a$ be the area associated with $v$ and $mnv$ be the current mean number of visits over all the areas; we choose to perform $v$ if and only if $nv_a \leq w_a \cdot mnv$ *i.e.*, if and only if the number of visits of $a$ is less than or equal to the mean number weighted by $a$'s weight;
- for the choice of having or not the instrument ON between $t$ and the beginning $t'$ of $v$, we choose to have it ON if and only if $t' - t \leq Dm$ (not too long ON time because of energy, memory, and temperature constraints, with $Dm$ a parameter to be set);
- finally, for the choice of remaining on the left or coming back to the right at the end of $v$, when $v$ is a compulsory observation to be performed on the left (ending at time $t''$) and the next compulsory observation must be performed on the left too (starting at time $t'''$), we choose to remain on the left if and only if $t''' - t'' \leq 2 \cdot Dc + Dm'$ (with $Dc$ the duration of an attitude movement and $Dm'$ a parameter to be set); however, if it is decided to come back to the right, we choose to maintain the instrument ON during the attitude movement.

## A non chronological greedy algorithm

The second algorithm builds a plan in a non chronological way by starting from a plan that contains only the compulsory observations and by inserting in it area visits one after the other until no visit can be added.

The algorithm starts by building for each satellite a plan that includes only the compulsory observations (it is assumed that such a plan exists). To build it, it creates a small OBS interval associated with each compulsory observation. If two successive observations must be performed on the left, it decides on remaining on the left or coming back to the right between them. Necessary attitude movements are then added to the plan. If the maximum number of ON/OFFs over the planning horizon is exceeded, successive OBS intervals are merged until the constraint on the number of ON/OFFs be satisfied, while satisfying constraints on energy, memory, and temperature.

Then, the algorithm tries to add to this plan as many area visits as possible. To do that, it selects an area $a$, a satellite $s$, and a visit $v$ of $a$ by $s$ that is not covered yet by the current OBS intervals. It creates a small OBS interval associated with $v$ (if there is an intersection between $v$ and a current OBS interval $i$, $i$ is simply extended). As previously, if the maximum number of ON/OFFs over the planning horizon is exceeded, successive OBS intervals are merged. If merging does not succeed (too many remaining ON/OFFs), then $v$ is not added to the plan and is removed from the set of candidate area visits. The algorithm stops when this set is empty. See Figure 9 for an illustration.

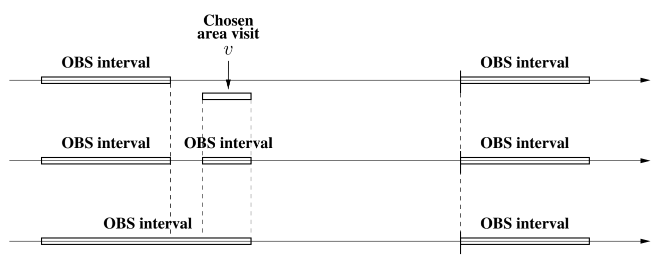


Figure 9: Choice of an area visit $v$, creation of an OBS interval, and fusion with another one.

As the previous algorithm, this algorithm contains some choice points at which local heuristics can be used in order to guide the search towards plans of reasonable quality. The following heuristics have been implemented and used in experiments:

- for the choice of the area $a$ for which a visit will be added to the plan, we choose an area that maximizes the immediate gain resulting from this addition *i.e.*, $w_a \cdot \alpha^{nv_a}$

(see Paragraph *Optimization criterion* in Section *Planning problem*);

- for the choice of the visit $v$ of $a$, we choose it randomly among the visits that are not covered yet by the current OBS intervals;
- finally, for the choice of the two successive OBS intervals to be merged, we choose the ones that are the closest from each other.

## Experimental results

### Toy instances

The two algorithms have been first experimented on very small artificial toy instances on which the OPL tool is able to produce optimal solutions and to prove optimality.

These instances involve 16 areas and 2 possible visits per area. They have been solved with different values of the satellite parameters: maximum consumption of energy and memory, maximum instrument temperature, and maximum number of ON/OFFs.

On these instances, the two algorithms produce very similar results, with the non chronological greedy algorithm being sometimes slightly more efficient than the chronological one (until 8.5% in terms of criterion value).

On the same instances, the comparison with the results produced by the OPL tool shows that the two algorithms produce most of the time optimal or quasi-optimal solutions, except when the temperature constraint is strong, with a maximum distance to the optimum of 20%.

### Real size instances

Real size instances have been solved with different satellite configurations and different satellite parameter values. Table 1 shows typical results we obtained.

| | Chronological greedy algorithm | Non chronological greedy algorithm |
|---|---|---|
| Criterion value | 41,837.25 | 41,925.40 |
| Number of area visits | 154,067 | 175,471 |
| Mean number of visits per area | 4.20 | 4.78 |
| Maximum number of visits per area | 17 | 28 |
| Minimum number of visits per area | 0 | 1 |
| Number of non visited areas | 1 | 0 |

Table 1: Experimental results on real size instances.

These results show that the non chronological greedy algorithm is clearly more efficient than the chronological one: more area visits are performed, the mean and the maximum number of visits per area are greater, and all the areas are visited at least once (with the chronological algorithm, one area is not visited at all). The small difference between both algorithms in terms of criterion value is due to the form of the criterion: when, for a given area $a$, a significant number of visits is already planned, adding one returns almost nothing ($w_a \cdot (1 - \alpha) \cdot \alpha^{nv_a}$, where $nv_a$ is the number of visits already planned).

Independently of the specific heuristics used in both algorithms, we guess that such a difference can be explained by the fact that the non chronological algorithm is less blind than the chronological one, which sees nothing beyond the next area visit it chooses, except checking that all the remaining compulsory observations can be still performed.

However, to be fair, it must be stressed that the non chronological greedy algorithm is a bit more complex to be implemented and less efficient in terms of computing time, mainly because of insertions of visits anywhere in the current plan, which require recomputing complex state evolutions such as the instrument temperature profile. On the contrary, with the chronological algorithm, the system state can be updated at each step in an incremental way.

## Conclusion and future work

These first results show that at least greedy algorithms can be practically used to solve the very large daily instances and to obtain plans of reasonable quality.

However, several directions could be explored in the next months.

From the algorithmic point of view, other heuristics and other heuristic parameters could be designed and experimented for both algorithms. Moreover, the non chronological algorithm can be seen as the first phase of a local search algorithm which would alternates visit additions and removals and would be equipped with simulated annealing and tabu search capabilities (Aarts and Lenstra 1997).

From the problem modeling point of view, other features should be taken into account in the constraints or in the criterion, such as a minimum number of visits per area, depending on the area, and an as regular as possible spreading of these visits over the planning horizon.

Not only acquisition, but also data downloading, should be taken into account in the planning process, and the age of the information (difference between the time at which acquisition is performed and the time at which associated data is delivered to the ground) should be considered.

Finally, algorithms should be able to behave in a repair mode in order to accept last minute high priority observations by disturbing as less as possible the current plan.

## References

[Aarts and Lenstra 1997] Aarts, E., and Lenstra, J., eds. 1997. *Local Search in Combinatorial Optimization*. John Wiley & Sons.

[Bellman 1957] Bellman, R. 1957. *Dynamic Programming*. Princeton University Press.

[IBM ILOG ] IBM ILOG. Cplex optimization studio. www-01.ibm.com/software/integration/optimization/cplex-optimization-studio/.

[Verfaillie and Lemaître 2006] Verfaillie, G., and Lemaître, M. 2006. Planning Activities for Earth Watching and Observation Satellites and Constellations: From Off-line Ground Planning to On-line On-board Planning. Tutorial given at ICAPS-06 (16th International Conference on Automated Planning and Scheduling, Cumbria, UK), www.plg.inf.uc3m.es/icaps06/preprints/i06-tu5-allpapers.pdf.