Building a really executable plan for a constellation of agile Earth observation satellites

Romain Grasset-Bourdel[⋆†] and **Gérard Verfaillie**[⋆]
[⋆] Onera - The French Aerospace Lab, Toulouse, France
{Romain.Grasset, Gerard.Verfaillie}@onera.fr

Antoine Flipo[†]
[†] CNES, Toulouse, France
Antoine.Flipo@cnes.fr

Abstract

In this paper, we present the work that has been done for the automatic planning of activities of a constellation of next-generation agile Earth-observing satellites. With regard to previous works about the management of similar satellites, the main result of this work is a chronological forward backtrack search algorithm which is able to take into account in an integrated way all the satellite activities (orbital manoeuvres, observations, data downloads, geocentric or heliocentric pointings, attitude movements, instrument activations . . .) and which guarantees the production of a plan that may be not optimal, but satisfies all the physical constraints and is therefore really executable by the satellites.

Introduction

The context of the work we present in this paper is the European defence MUSIS project (Multinational Space-based Imaging System for Surveillance, Reconnaissance, and Observation) and more precisely the management of the MUSIS agile satellites that are equipped with high-resolution optical observation instruments.

As usual, such satellites are managed from the ground by a mission planning system which receives user observation requests, builds regularly satellite activity plans over a limited horizon ahead (typically one day), and receives plan execution reports. These plans must meet all the physical constraints and satisfy as well as possible the user requests.

The planning algorithms that have been designed and implemented so far for such satellites (see for example (Lemaître et al. 2002)) build first an observation plan, taking into account visibility windows, observation durations, and transition durations between observations. Then, a data download plan is built, taking into account communication windows and download durations. Finally energy and memory constraints are checked. In case of constraint violation, downloads or observations are removed until constraint satisfaction is restored, leading potentially to strongly sub-optimal solutions.

One of the challenges of the work we present in this paper was to design algorithms able to build in a single pass a plan that covers all the satellite activities (orbital manoeuvres, observations, data downloads, geocentric or heliocentric pointings, attitude movements, instrument activations . . .), meets all the physical constraints, satisfies as well as possible the user requests, and requires a limited computing time (typically up to ten minutes).

The result is a chronological forward search algorithm with dedicated decision heuristics, constraint checking, limited lookahead, and backtrack in case of constraint violation, which guarantees the production of a plan that may be not optimal, but is really executable by the satellites.

In this paper, we present the physical system we have to manage, the physical constraints, the user requests, and the organization of the management system we assume. We analyze the resulting planning problem with regard to classical planning and scheduling problems. Then, we describe the dedicated algorithm we designed. Finally, we report experimental results.

Satellite constellation

The constellation we consider is made up of two identical satellites[1] moving on the same orbit (circular, low altitude, quasi-polar, and heliosynchronous) with a phase shift of 180 degrees between the two satellites (see Figure 1).

Each satellite (see Figure 2) is equipped with thrusters which allow orbital manoeuvres to be performed in case of a too important drift with regard to the reference orbit and with gyroscopic actuators which allow very quick attitude movements (agility) useful to perform observations and transitions between observations.

A telescope, with two focal planes, allows observations to be performed in the visible and infra-red spectra, with two images (visible and infra-red) within day periods (on the ground) and only one image (infra-red) within night periods. A mass memory allows observation data to be recorded and a high-rate large-aperture antenna allows it to be downloaded towards ground reception stations. Solar panels allow batteries to be recharged when the satellite is not in eclipse. For the sake of agility, all these equipments are body-mounted on the satellite.

[1] The planning algorithm we propose is able to manage any number of satellites, possibly not identical: not the same parameter values.

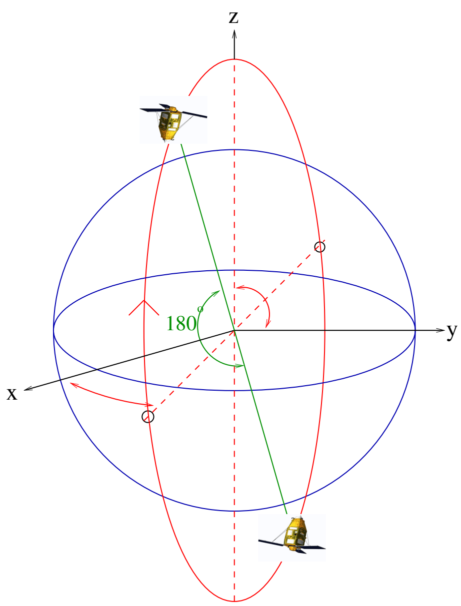

Figure 1: Two-satellite constellation.

Figure 2: Artist view of a satellite.

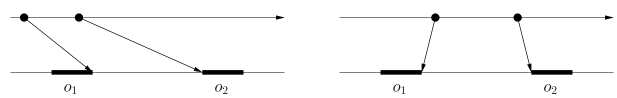

Figure 3: How the angular distance and thus the minimum transition time between observations depends on the time at which the first ends.

Physical constraints

The physical constraints that must be met can be classified into six classes : attitude trajectory, observation, download, memory, instruments, and energy.

Attitude trajectory Thanks to gyroscopic actuators, the satellite is permanently moving around its gravity centre along the three axes (roll, pitch, and yaw). These attitude movements allow observations of areas on the ground to be performed by scanning them. They also allow transitions from the end of an observation to the beginning of the following to be performed relatively quickly. These movements are limited in terms of angular speed and acceleration, resulting in minimum times for moving from an attitude to another. However, the attitude that is required to observe a given area on the ground depends on the orbital position of the satellite and thus on the time at which the observation is performed. The result is a minimum time between the end of an observation and the beginning of the following that depends on the time at which the first ends (see Figure 3 for a schematic 2D illustration). Moreover, computing this minimum time requires solving a complex continuous optimization problem (see (Beaumet, Verfaillie, and Charmeau 2007)). For solving it efficiently inside planning algorithms, dedicated approximate algorithms have been developed at ONERA, assuming three-phase movements (constant acceleration, constant speed, and constant deceleration) performed concurrently along each axis (roll, pitch, and yaw).

Observation Due to maximum observation angles, the observation of a given area on the ground must be performed within one of its visibility windows. Its duration is fixed, because it only depends on the required scanning speed on the ground. The satellite attitude trajectory to be followed during observation depends on the time at which it starts.

Download The same way, due to maximum communication angles, a data download must be performed within one of the visibility windows of one of the ground reception stations. However, this does not suffice because the satellite attitude must be compatible with download (the target station must remain within the satellite antenna communication cone). The result is a set of effective communication windows that depends on the satellite attitude trajectory. Observation and download can be performed concurrently. Two images (visible and infra-red) resulting from the same observation (within a day period) must be downloaded towards the same station and during the same station overflight.

Memory The amount of memory available on board for observation data recording must be never exceeded.

Instruments Concerning the three instruments (high-rate antenna, visible and infra-red focal planes) a minimum preheating time must be met before use, as well as a maximum total ON time and a maximum number of ON/OFF cycles over the planning horizon, for the sake of reliability. Temperature of the infra-red focal plane is automatically regulated by a cryothermic system, but temperatures of both the visible focal plane and the antenna must remain below a given level. Moreover, it must be checked that, at every point on the satellite attitude trajectory, the focal planes are not dazzled and thus not damaged by the sunlight (minimum angle between the satellite axis and the Sun direction).

Energy On-board energy cannot exceed a maximum level due to battery limitations. For the sake of safety, it must remain above a given level, particularly when the satellite is in eclipse and solar panels produce nothing. When the satellite is not in eclipse, the production of energy via the solar panels depends on the satellite attitude trajectory. On the other hand, energy consumption depends on instrument activations.

User requests

With each user request, are associated a polygon which has been split into strips, a priority level, a weight, and a deadline. Typically, three priority levels are available, from 3 (the highest) to 1 (the lowest). It is assumed that any request of priority p is preferred to any set of requests of priority strictly less than p. Weights allow to express preferences between requests of the same priority level and are assumed to be additive. In general, user requests exceed the constellation capacity and choices must be made using request priorities and weights.

It is assumed that any strip can be observed using only one strip overflight. With each strip, are associated a geographical definition, observation durations (day or night), image sizes (visible, day or night infra-red), a maximum observation angle, and a set of triples ⟨satellite, visibility window, weather forecast⟩.

Management system

For the moment, it is assumed that user requests arrive at any time and that, each day, at a given time, a plan is built for the next day from all the requests that are not out of date and not fully satisfied yet. This plan is built on the ground and then uploaded to the satellites for execution. Analysis of observation data, taking into account the actual cloud cover, allows satisfied requests to be removed.

In the future, a more flexible management mode will be considered, allowing urgent requests to be quickly taken into account at any time of the day and using several plan uploading opportunities from several satellite control stations. This is out of the scope of this paper.

Problem modeling

This planning problem can be modeled using for each satellite the following state variables:

- the current time and thus the orbital position;
- the attitude position and speed along the three axes;
- the available memory and energy;
- for each instrument, its status (ON or OFF), the remaining ON time, and the remaining number of ON/OFF cycles;
- for the antenna and the visible focal plane, its temperature.

Six types of action are available for each satellite:

1. orbital manoeuvres which are mandatory and characterized by starting and ending times and attitudes and by an energy production (function of the attitude trajectory during the manoeuvre);
2. observations which are characterized by a strip, a visibility window, and a starting time;
3. data downloads which are characterized by an image, a reception station, a communication window, and a starting time;
4. heliocentric pointings (solar panels directed towards the Sun in order to recharge batteries as fast as possible) which are characterized by starting and ending times;
5. geocentric pointings (satellite axis directed towards the Earth centre; default action when there is nothing else to do) which are characterized by starting and ending times too;
6. instrument switchings which are characterized by an instrument and a time.

It must be observed that actions of all the types, but the third and sixth (data downloads and instrument switchings), constrain the satellite attitude and are thus mutually exclusive. They must be performed in sequence. Only data downloads and instrument switchings can be performed in parallel, at any time for instrument switchings, but only within effective communication windows for data downloads. As a consequence, a plan has the form of a sequence of actions of any type, except the third and sixth, with attitude movements between consecutive actions and with data downloads and instrument switchings in parallel.

Any plan must satisfy all the constraints described above in Section *Physical constraints*.

The criterion to be optimized is a vector of numbers v_p, one for each priority level p. Two vectors resulting from two plans are lexicographically compared. For each priority level p, v_p is the sum of the weights of the requests r of priority p, weighted by four factors whose value is between 0 and 1 and which represent (1) the percentage of realization (observation and data download), (2) the mean percentage of cloud cover, (3) the mean observation angle, and (4) the mean data delivering delay, over all the strips of the polygon associated with r.

Problem analysis

The resulting planning problem has some connections with the well-known *knapsack* problem (Kellerer, Pferschy, and Pisinger 2004) if we disregard the temporal aspects and consider that observations are objects and satellites are sacks. Particularly, the data download subproblem is very close to the academic knapsack problem.

This planning problem has also obvious connections with *scheduling* problems if we consider that observations are tasks and more particularly with *oversubscribed scheduling* problems (Barbulescu et al. 2006; Kramer, Barbulescu, and Smith 2007) where not all the tasks must be scheduled, but a function of the scheduled tasks must be optimized.

One of the distinctive features of the problem we face is however the existence of minimum transition times between observations which not only depend on the sequence of observations, as in *scheduling with sequence-dependent setup times*, but depend on the times at which observations end, more or less as in *time-dependent scheduling* (Cheng, Ding, and Lin 2004; Gawiejnowicz 2008). However, in time-dependent scheduling problems, task durations depend on time whereas, in the problem we face, minimum transition times depend on time. More importantly, in time-dependent scheduling problems, it is assumed that task durations are given by an analytical formula, often a linear or piecewise linear function of time, whereas, in the problem we face, minimum transition times are given by a computing code which approximates the optimum of a complex continuous optimization problem (no analytical formula available and no reasonable linear approximation).

Moreover, in our planning problem, the energy that is produced by the solar panels, the effective communication windows, and the absence of focal plane dazzle depend on the actual satellite attitude trajectory, which is computed by dedicated algorithms from the choices made in terms of orbital manoeuvres, observations and pointings (geo or heliocentric). Some of the constraints that we must take into account are similar, in terms of complexity of modeling and checking, to the thermal and pointing constraints taken into account in (Chien et al. 2010) for scheduling operations on board the EO-1 satellite.

The problem we face mixes features from AI and OR planning and scheduling (Ghallab, Nau, and Traverso 2004) with other features from motion planning (LaValle 2006). This justifies the development of a dedicated planning algorithm, even if some heuristics and bounds developed to solve the problems cited above can be profitably used in this algorithm (see in the next section the use of heuristics inspired from usual knapsack heuristics or the propagation of latest observation ending times inspired from scheduling propagation mechanisms).

Although the problem may seem to be similar, we did not use the algorithms presented in (Beaumet, Verfaillie, and Charmeau 2011) because of too important differences in the problem definition. In the problem we face, instrument dazzle, activation, and temperature are considered, and data downloads are allowed concurrently with observations. Moreover, in (Beaumet, Verfaillie, and Charmeau 2011), the objective was to build plans over a short planning horizon ahead in order to make right decisions on board. Here, the problem is to build on the ground over a one-day horizon plans that are really executable by the satellites and thus take into account precisely all the physical constraints.

Algorithm

Decreasing priorities First, the algorithm we developed works by decreasing priority levels from 3 (the highest) to 1 (the lowest). At each priority level p, the starting point is the plan Pl produced at the previous level $p+1$, which includes orbital manoeuvres, observations (of priority $p+1$ or more), pointings (geo or heliocentric), data downloads, and instrument switchings. However, what is kept from Pl is only the sequence Seq of orbital manoeuvres and observations present in Pl, without their starting times. Other actions present in Pl, such as pointings, data downloads, or instrument switchings are disregarded. At level p, observations of priority p will be inserted into Seq by moving starting times when necessary. Other actions, such as pointings, data downloads, or instrument switchings will be added to build a consistent plan. At priority level 3, the starting point is the set of orbital manoeuvres which are imposed on the mission planning system by the satellite control system and can be classed as observations of priority 4.

Such an approach is justified by the fact that any request of priority strictly greater than p is preferred to any set of requests of priority p. This leads us to consider the sequence of observations present in the plan produced at level $p+1$ as being mandatory when building a plan at level p.

A forward chronological algorithm At each priority level p, the algorithm builds a plan in a forward chronological way, from the beginning Ts of the planning horizon to the end Te. With any step of the algorithm, are associated the current time t, the next observation o of priority $p+1$ or more to be included in the plan because it belongs to Seq, and the set Os of observations of priority p that can be scheduled after t and before o. At the first step, $t = Ts$ and o is the first observation in Seq. The algorithm chooses an observation o' in Os as the next observation to be included in the plan and a starting time t' for o'. If $Os = \emptyset$, then $o' = o$ (observation o is chosen and then a starting time for it). At the next step of the algorithm, t is replaced by the ending time t'' of o' and, if $o' = o$, then o is replaced by the observation that follows it in Seq (empty when o is the last observation in Seq). Figure 4 illustrates two successive steps of the algorithm. The algorithm stops when o and Os are both empty (no other observation to be included in the plan).

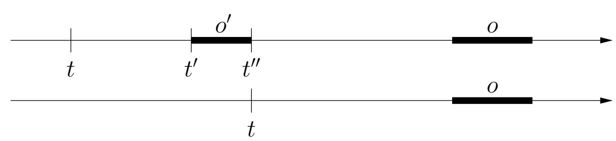

Figure 4: Two successive steps of the forward chronological algorithm.

Decision levels This is the first decision level (1) of the algorithm (choice of the next observation to be included). Once this choice has been made, the algorithm makes other choices over the temporal horizon from t to t'' at other decision levels: (2) possible insertion of geo or heliocentric pointings, (3) possible data downloads, and (4) instrument activations.

At the second decision level, geo or heliocentric pointings are inserted between t and t' when possible. Once insertions have been decided, the satellite attitude trajectory is completely fixed from t to t''. Hence, the production of energy and the effective communication windows can be computed and the absence of focal plane dazzle can be checked by simulating trajectories.

At the third decision level, data downloads are inserted within the effective communication windows from t to t'' and memory constraints can be checked. This means that observations (first decision level) have priority over downloads (third level). This choice is justified by mission and algorithm considerations: on the one hand, observation is the main system bottleneck and, on the other hand, it is necessary to know the effective communication windows and thus observations and pointings before planning downloads.

At the fourth decision level, instrument activations are inserted in order to satisfy the requirements in terms of observation (visible and infra-red focal planes) and download (high-rate antenna). Energy and instrument constraints can be checked.

Figure 5 shows an example of decisions at the four levels: at level 1, observation o', starting at t', is chosen; at level 2, a geocentric pointing followed by a heliocentric one are inserted before t'; at level 3, data downloads d_1 and d_2, followed by d_3 and d_4, are inserted between t and t''; at level 4, the following decisions are made concerning instrument activations: at time t, the visible focal plane was OFF and it is decided to switch it ON only before o'; on the contrary, the infra-red focal plane was ON and it is decided to maintain it ON between t and t''; at time t, the antenna was OFF, and it is decided to switch it ON before downloading d_1 and to maintain it ON until the end of d_4's download.

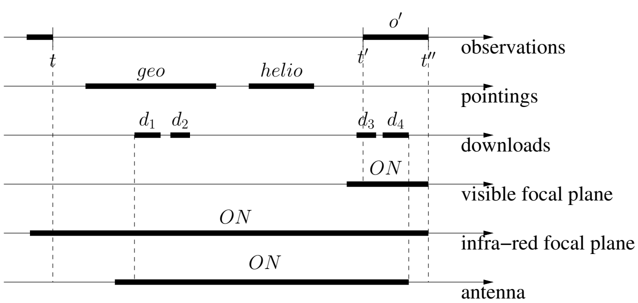

Figure 5: Example of decisions at the four levels: (1) observations, (2) pointings, (3) downloads, and (4) instruments.

Once decisions have been made at the four levels, a consistent plan is available from t to t'', extending the plan that already exists from Ts to t, and the planning process can continue from t'', starting from a completely known satellite state.

This incremental process, which built incrementally a complex system trajectory, is the main justification for using a forward chronological search.

For the sake of simplicity, we present the algorithm by assuming only one satellite. However the planning process is in fact interleaved on the two satellites and the next planning step is the earliest one over the two satellites.

Backtracks At any decision level, in case of constraint violation, other choices are made. If no other choice is available, a hierarchical backtrack at the relevant decision level is triggered. Figure 6 shows the hierarchical backtrack mechanisms beteeen decision levels. For example, if, at the fourth level (instruments), no consistent activation plan can be built for the antenna, a backtrack at the third level (downloads) is triggered but, if no consistent activation plan can be built for one of the focal planes, a backtrack at the first level (observations) is triggered.

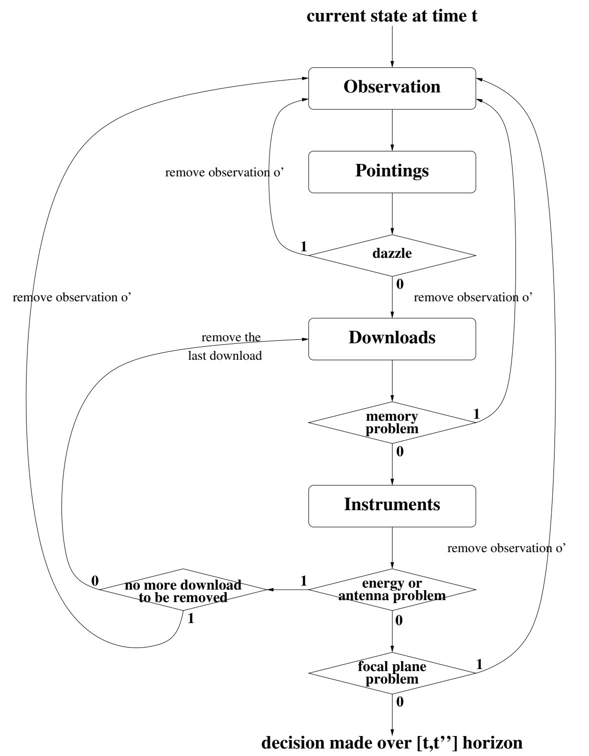

Figure 6: Hierarchical backtracks between decision levels.

At the first level, if $Os = \emptyset$ and thus o is chosen, but insertion of o is impossible, a chronological backtrack is triggered to the previous insertion of an observation of priority p. However, in order to avoid as much as possible such situations, the latest observation ending times are propagated from the end to the beginning of Seq before planning.

Heuristics At all the decision levels, heuristics are necessary to make choices. These heuristics are crucial to the production of good quality plans because, for the sake of efficiency, the algorithm backtracks only in case of constraint violation and never to try and improve on the current plan. It may be important to stress the difference between the global optimization criterion defined in Section *Problem modeling* and the local heuristics described below which only aim at guiding the search towards good quality solutions.

The following heuristics have been implemented at the various decision levels:

1. at the first level, as in the knapsack problem, one chooses an observation o' that maximizes the ratio between the increase in the criterion resulting from the insertion of o' (gain) and the time consumed by this insertion ($t'' - t$, considering the earliest starting time for o'; cost); once an

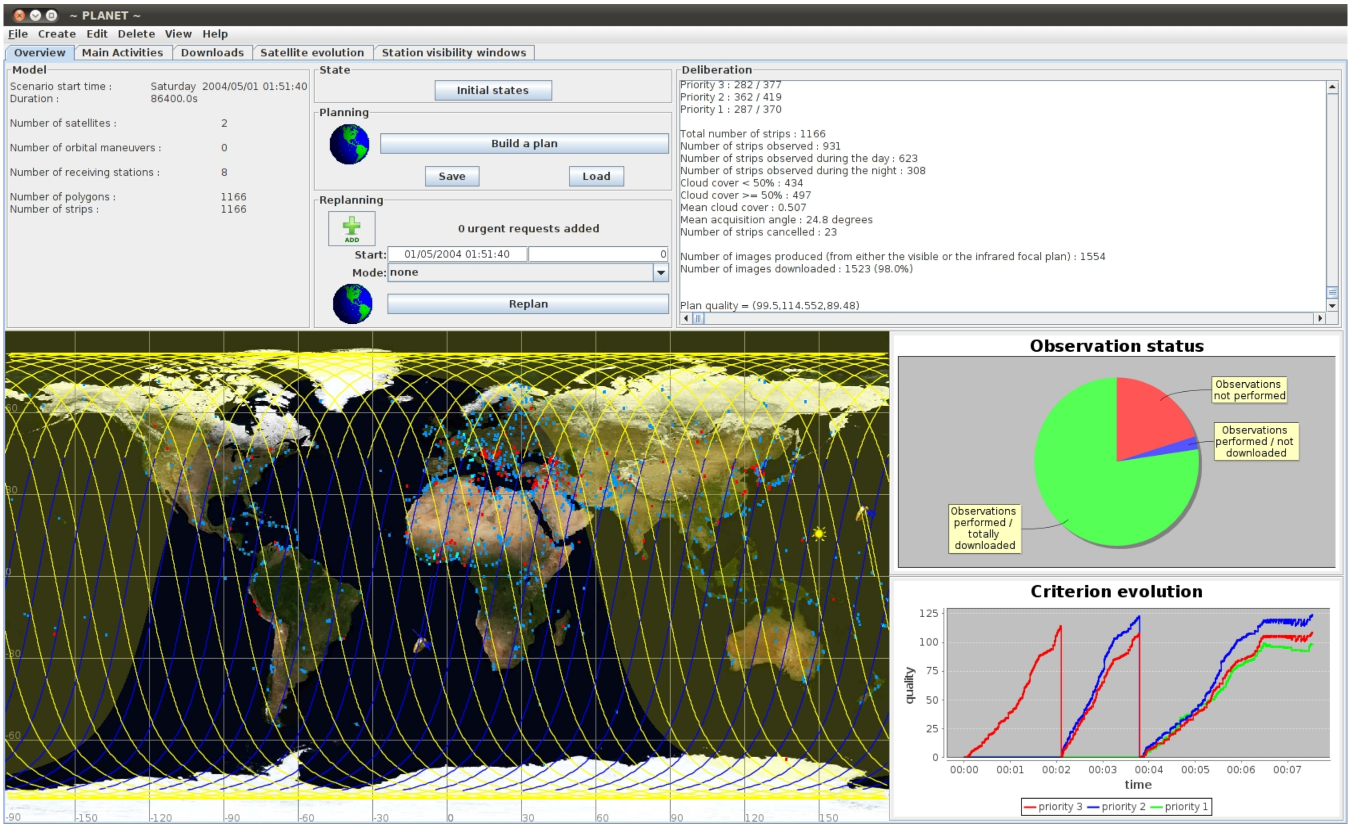

Figure 7: Top level interface of the PLANET tool when planning is complete.

observation o' has been chosen, one chooses for it a starting time t' that maximizes the increase in the criterion resulting from the insertion of o' at time t' (function of the observation angle; gain) minus the sum of the decreases in the criterion resulting from this insertion (other observations that would become impossible and whose quality would degrade because of too large observation angles; cost);

2. at the second level, an expert rule aims at making easier energy production and data download; it systematically chooses a geocentric pointing when the satellite is in eclipse; when it is not in eclipse, it gives priority to a heliocentric pointing in order to recharge batteries, except in case of visibility of a ground reception station, because a geocentric pointing always allows data download, whereas a heliocentric one may prevent it;

3. at the third level, as in the knapsack problem, one chooses an image of maximum priority that maximizes the ratio between the increase in the criterion resulting from its download (gain) and the duration of this download (cost);

4. at the fourth level, the choice is, for each instrument, at the end of each mandatory activity period, between switching it OFF and maintaining it ON; these choices have an impact on four "resources": energy, temperature, total ON time, and number of ON/OFF cycles; the result is a kind of multi-criteria decision problem; for each resource and for each alternative a, it is possible to compute a ratio between remaining and maximum quantity, if a is chosen; finally, as usual in multi-criteria decision making, one chooses the alternative that maximizes the minimum ratio over the four resources.

The main difference between this algorithm and the one presented in (Beaumet, Verfaillie, and Charmeau 2011) is the use of backtrack mechanisms in case of constraint violation and of more sophisticated choice heuristics. In (Beaumet, Verfaillie, and Charmeau 2011), no backtrack was allowed and heuristics were limited to randomized decision rules.

Planning tool

This algorithm is at the core of the experimental planning tool, called PLANET for *PLanner for Agile observatioN satElliTes*, which has been developed for this mission, on the basis of a previous tool (Beaumet, Verfaillie, and Charmeau 2011).

In addition to a planning algorithm, PLANET includes tools for specifying physical system parameters and user requests, for setting algorithm parameters, and for visualizing planning results, including the timelines that represent the evolution of the state of the satellites over the planning horizon.

Scenarios and experimental results

The evaluation of the proposed algorithm and its comparison with other ones is problematic because existing algorithms (see for example (Lemaître et al. 2002; Cordeau and Laporte 2005; Bianchessi et al. 2007; Beaumet, Verfaillie, and Charmeau 2011)) have been developed for other physical settings and cannot solve the problem we face today.

However, we were able to run the PLANET algorithm on a real-size realistic instance, built by CNES (French Space Agency) and whose characteristics are the following ones:

- a one-day planning horizon;
- 8 ground reception stations;
- 3 priority levels;
- 1166 observation requests, all of them with polygons limited to one strip and all of them of the same weight(1); among them, 377 of priority 3 (the highest), 419 of priority 2, and 370 of priority 1 (the smallest);
- meteorological forecast built from climatological data.

On this instance, planning takes a bit more than 7 minutes, using a 3Ghz Intel processor with 2.5Go of RAM, running under Linux. In the resulting plan, 906 (78%) observations are performed and downloaded, 16 (1%) are performed, but not downloaded, and 244 (21%) not performed at all. Among the observations of priority 3, 280 (74%) are performed. Results are 367 (88%) for priority 2 and 275 (74%) for priority 1. The fact that relatively more observations of priority 2 are performed than observations of priority 3 can be explained by the fact that, in this instance, observations of priority 3 are more geographically conflicting with each other. Moreover, due to the heuristics used at the first level, the algorithm prefers not to perform observations of low utility because of too bad meteorological forecast.

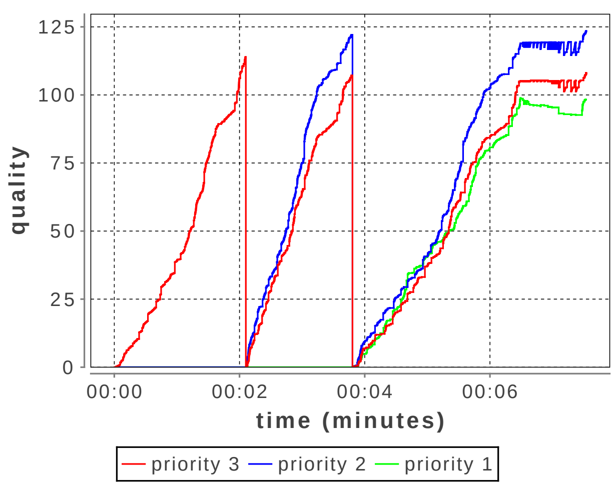

Figure 8: Evolution of the value of the current plan during the planning process.

Figure 7 shows the top level interface of the PLANET tool when planning on this instance is complete. Observations that are performed appear in blue on the planisphere, whereas those that are not performed appear in red. In the bottom right corner of the interface (focus shown in Figure 8), one can see how the value of the current plan evolves during the planning process. One can observe the three phases of this process, each one associated with a priority level: the first one with only observations of priority 3 (in red), the second one with observations of priority 2 and 3 (in blue for the former), and the third one with observations of priority 1, 2, and 3 (in green for the former). Alternations of increases and decreases in the criterion at the end of the third phase is due to chronological backtracks following constraint violations on observations of priority 2 or 3 (mandatory in this phase).

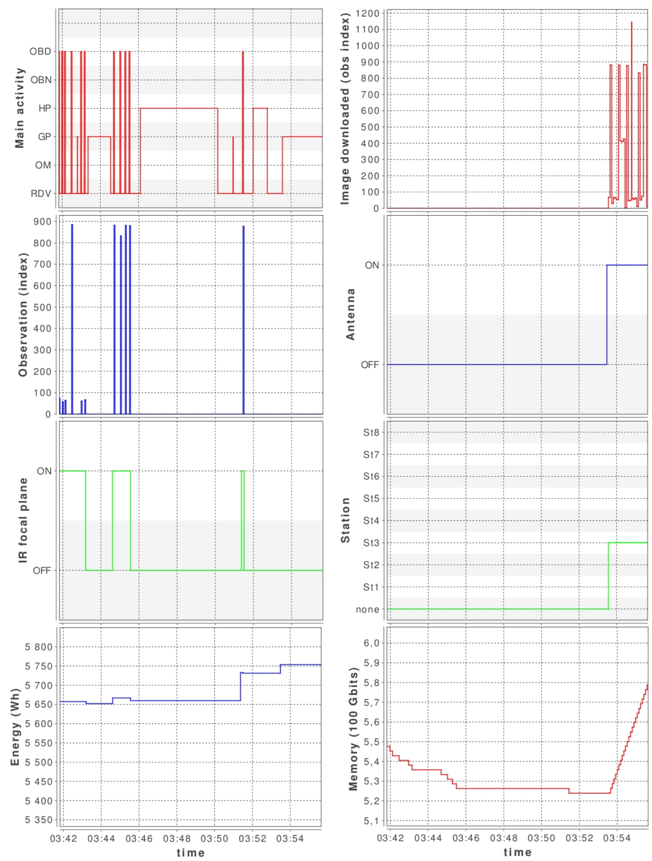

Figure 9: Focus on some timelines that represent the evolution of the state of one of the satellites, over a 14 minute horizon.

Figure 9 is a focus on some timelines that represent the evolution of the state of one of the satellites, over a 14 minute horizon. In the first column, the first timeline shows the main activity on board (OBN for night observation, OBD for day observation, HP for heliocentric pointing, GP for geocentric pointing, OM for orbital manoeuvre, and RDV for attitude rendezvous). Alternation of day observations, geo and heliocentric pointings, and attitude rendezvous can be observed. The timeline below shows, in case of observation, the index of the observation performed. The one still below shows ON/OFF cycles of the infra-red focal plane, which remains ON for the first five observations, for the following four, and then for the last. In the second column, the first timeline shows, in case of download, the index of the observation associated with the image downloaded. The timeline below shows ON/OFF cycles of the high-rate antenna, which remains ON for all the downloads. The one still below shows the ground station (in this case, always the third) towards which downloads are performed. The bottom two timelines show the evolution of energy and memory. For example, memory available on board decreases with observations at the beginning of the window and increases with downloads at the end.

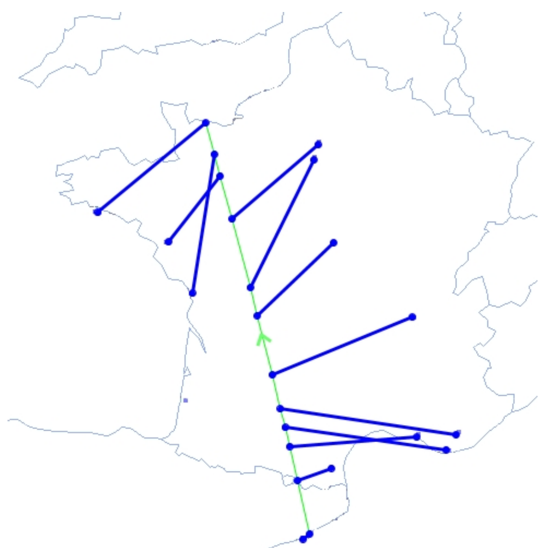

Figure 10: Plan produced by the PLANET algorithm.

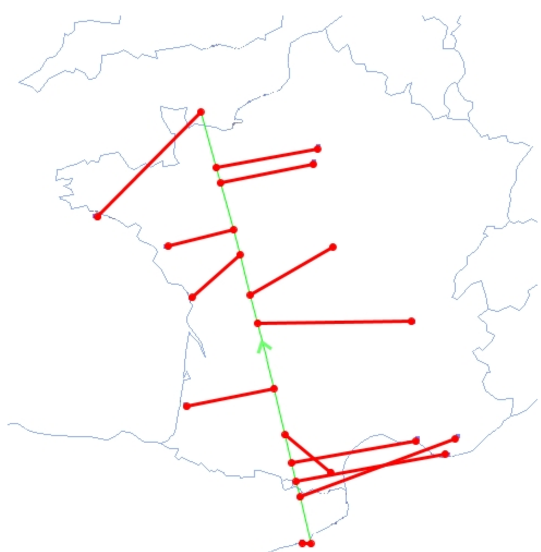

Figure 11: Optimal plan produced by an A^* algorithm.

In addition to these real-size experiments, we run the PLANET algorithm on a small but very constrained instance, involving 13 observations of areas that are geographically very close to each other. Each request has the same priority (3) and the same weight (1). Each area is limited to one strip. Meteorological forecast is the same (very good) for each of them.

On this instance, we compared the PLANET algorithm with an optimal A^* algorithm using a time discretization of one second. The PLANET algorithm is able to plan 12 observations with a criterion value of 9.48, whereas the A^* algorithm is able to plan all of them (13) with a criterion value of 10.58. Criterion values take into account only acquisition angles. Figures 10 and 11 show the plans produced by both algorithms. The ground track of the satellite flying over France from south to north is displayed in green. For each area, its short observation window and the projection of the satellite observation attitude, are displayed in blue or red. It must be however stressed that the PLANET algorithm produces this result in 2 seconds whereas the A^* algorithm takes 5241 seconds. Moreover, the A^* algorithm is unable to solve instances beyond 13 observations due to lack of RAM.

Conclusion

We think that the main result of this work is to show that is possible to build a planning algorithm which (i) is able to handle all the complex physical constraints present in the problem at hand, including attitude trajectory, observation, download, memory, instrument, and energy constraints, (ii) guarantees the production of a plan that may be not optimal, but is really executable thanks to constraint checking, and (iii) is able to produce in some minutes, over a one-day planning horizon, a plan with hundreds of observations and downloads, which covers satellite attitude trajectory as well as observation, data download, satellite pointing, and instrument activation activities.

The current work consists in adapting this algorithm to run in a repair mode. Even if a plan is built every day, it may be interesting to profit from the presence of several satellite control stations and from the associated plan uploading opportunities, and to modify during the day the remaining part of the plan, following new urgent observation requests or new meteorological forecast. To do that, the technical approach would be to use the same algorithm, by modifying the optimization criterion and the associated heuristics (in order to favour plan stability), and by running the algorithm in several successive modes, with a less and less restrictive search (in order to get an anytime behaviour; first solution produced very quickly and improved as far as computing time is available).

References

[Barbulescu et al. 2006] Barbulescu, L.; Howe, A.; Whitley, L.; and Roberts, M. 2006. Understanding Algorithm Performance on an Oversubscribed Scheduling Application. *Journal of Artificial Intelligence Research* 27:577–615.

[Beaumet, Verfaillie, and Charmeau 2007] Beaumet, G.; Verfaillie, G.; and Charmeau, M. 2007. Estimation of the Minimal Duration of an Attitude Change for an Autonomous Agile Earth-observing Satellite. In *Proc. of the 13th International Conference on Principles and Practice of Constraint Programming (CP-07)*, 3–17.

[Beaumet, Verfaillie, and Charmeau 2011] Beaumet, G.; Verfaillie, G.; and Charmeau, M. 2011. Feasibility of Autonomous Decision Making on board an Agile Earth-observing Satellite. *Computational Intelligence* 27(1):123–139.

[Bianchessi et al. 2007] Bianchessi, N.; Cordeau, J.; 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.

[Cheng, Ding, and Lin 2004] Cheng, T.; Ding, Q.; and Lin, B. 2004. A Concise Survey of Scheduling with Time-dependent Processing Times. *European Journal of Operational Research* 152:1–13.

[Chien et al. 2010] Chien, S.; Tran, D.; Rabideau, G.; Schaffer, S.; Mandl, D.; and Frye, S. 2010. Timeline-based Space Operations Scheduling with External Constraints. In *Proc. of the 20th International Conference on Automated Planning and Scheduling (ICAPS-10)*.

[Cordeau and Laporte 2005] Cordeau, J., and Laporte, G. 2005. Maximizing the Value of an Earth Observation Satellite Orbit. *Journal of the Operational Research Society* 56(8):962–968.

[Gawiejnowicz 2008] Gawiejnowicz, S. 2008. *Time-dependent Scheduling*. Springer.

[Ghallab, Nau, and Traverso 2004] Ghallab, M.; Nau, D.; and Traverso, P. 2004. *Automated Planning: Theory and Practice*. Morgan Kaufmann.

[Kellerer, Pferschy, and Pisinger 2004] Kellerer, H.; Pferschy, U.; and Pisinger, D. 2004. *Knapsack Problems*. Springer.

[Kramer, Barbulescu, and Smith 2007] Kramer, L.; Barbulescu, L.; and Smith, S. 2007. Understanding Performance Tradeoffs in Algorithms for Solving Oversubscribed Scheduling. In *Proc. of the 22nd National Conference on Artificial Intelligence (AAAI-07)*, 1019–1024.

[LaValle 2006] LaValle, S. 2006. *Planning Algorithms*. Cambridge University Press.

[Lemaître et al. 2002] Lemaître, M.; Verfaillie, G.; Jouhaud, F.; Lachiver, J.-M.; and Bataille, N. 2002. Selecting and scheduling observations of agile satellites. *Aerospace Science and Technology* 6:367–381.