Designing and Evaluating an On-line On-board Autonomous Earth Observation Satellite Scheduling System

Gérard Verfaillie
ONERA/DCSD, Toulouse Center
2, av. Édouard Belin, BP 4025
31055 Toulouse Cedex 4, France
phone: +33 5 62 25 26 32
fax: +33 5 62 25 25 64
email: verfaillie@cert.fr

Éric Bornschlegl
European Space Agency, ESA/ESTEC/ESD
Keplerlaan 1
2201 AG Noordwijk, The Netherlands
phone: +31 71 565 34 87
fax: +31 71 565 42 95
email: ebornsch@estec.esa.nl

Abstract

Managing an Earth observation satellite consists in building at regular intervals a feasible optimal sequence of image acquisitions over a given time horizon. Up to now this management was performed off-line and on the ground by semi-automated mission control centers, under the supervision of human operators. But the next generation satellites shall be managed autonomously on-line and on-board.

In this paper, after some background about Earth observation satellites and their management, we present the potential advantages and drawbacks of an evolution towards an on-line and on-board management. Then, we present two versions of a scheduling algorithm, both based on *Dynamic programming*: the first dedicated to off-line management and the second to on-line management. Finally experiments, using simplified models, show that an on-line management, performed on-board and using consequently limited computing resources, may outperform an off-line management, performed on the ground.

Background about Earth observation satellites

Orbit features

Earth observation satellites use circular polar sun-synchronous phased orbits. Combined with the natural Earth rotation, this kind of orbit allows any point of the Earth surface to be flown over several times during the orbit cycle under the same altitude and illumination conditions.

Depending on the orbit parameters, a revolution around the Earth takes about one hour and a cycle about ten days. Thus, depending on the latitude, the time between two successive flights over a given point of the Earth surface ranges between several hours and several days.

Image acquisition

Depending on the mission, satellites are equipped with various observation instruments (either optical or infrared instruments, radars ...). These instruments generally offer various spectral bands and acquisition modes. With an optical instrument, acquisition is possible only during the illuminated part of each revolution.

Image acquisition on both sides of the satellite ground track is generally possible using either mobile mirrors in front each instrument, or satellite manoeuvering ability. Satellites of the next generation will offer manoeuvering ability according the roll, pitch and yaw axes, and thus the maximal flexibility for image acquisition.

Data delivery

Data associated with an image acquisition can be, either directly down-loaded, when the satellite is within the visibility window of a ground station, or recorded on-board and down-loaded within one of the next ground station visibility windows.

Uncertainty about accomplishment

With an optical instrument, the possible presence of clouds may invalidate the image acquisition. In that case, the image has to be acquired another time, using other acquisition opportunities.

User requests

Users order image acquisitions, with associated requirements about the acquisition conditions: possible modes and spectral bands, minimum and maximum pitch and roll angles, minimum and maximum sun angle, either mono or stereo acquisition, time window, possible cyclic acquisition and associated constraints, grounds stations to which data may be down-loaded ... Priority levels or weights are generally associated with each user request.

Physical and technological constraints

The main physical constraint is associated with the instrument resources: at any time, only one image can be acquired using one instrument; moreover a transition time is generally necessary between two successive image acquisitions (changes in acquisition mode, in mirror orientation, or in satellite attitude).

The other constraints are associated with the energy and memory resources. Energy and memory are consumed during and between image acquisitions and are renewed via solar arrays and data down-loading. One of the ways of taking these constraints into account consists in defining for each of them a maximum consumption by revolution or set of revolutions, that guarantees that everything that is consumed will be renewed.

Managing an Earth observation satellite

Global view

Globally speaking, managing an Earth observation satellite consists in building over a given time horizon a sequence of image acquisitions that:

- satisfies all the physical and technological constraints associated with the satellite and its payload,
- satisfies the requirements associated with each user request,
- is optimal according to a criterion that is a function of the set of selected requests.

Scheduling horizon

The scheduling horizon can range from one part of a revolution to several revolutions over one or several days. Large horizons allow the global constraints such as energy or memory constraints, as well as the several accomplishment opportunities associated with each user request, to be better taken into account. But large horizons induce very large combinatorial problems and force to reason about to too uncertain data (accomplishment of selected requests, arrival of new requests ...).

Selection assessment criterion

Various assessment criteria may be used. The most used is utilitarist: a weight is associated with each user request and the objective is to maximize the sum of the weights of the selected requests. But an egalitarist criterion might be used: a priority level is associated with each user request and the objective is to minimize the maximum level of all the not selected requests. Subsequently, we will assume an utilitarist criterion.

A constraint optimization problem

Thus, managing an Earth observation satellite implies to solve a constrained optimization problem. When compared with classical *Operations Research* problems, this problem lies somewhere between the *Job-Shop Scheduling* problem (tasks to perform, not sharable instrument resources, transition times between tasks) and the *Multi-Knapsack* problem (cumulative energy and memory resources)

Potential representation frameworks

It can be cast in any of the classical frameworks used to represent constrained optimization problems, such as *Integer Linear Programming* (*ILP*; the criterion is linear and constraints can be linearized) or *Constraint Programming* (*CP*).

Potential solving methods

Any of the main associated methods can be used to solve it: either optimal methods like *Branch and Bound* or *Dynamic Programming*, or approximate methods like *Local Search* or *Greedy Algorithms*.

For example, the specific management problem of the French satellite *Spot5*, described in (BLV99), has been cast using the *ILP* and *CP* frameworks (*Cplex* and *Ilog Solver* tools), as well the *Valued Constraint Satisfaction Problem* framework (*VCSP*; an extension of the *CSP* framework to deal with over-constrained problems (SFV95)). It has been dealt with using associated *Branch and Bound* algorithms, as well as *Local Search* or *Greedy* algorithms.

Towards an on-line on-board management system

Current management systems

Earth observation satellites are currently managed on the ground and off-line: image acquisition and data downloading sequences over a given horizon are determined in the frame of semi-automated mission control centers, under the supervision of human operators and then up-loaded using satellite visibility windows.

On-line versus off-line management

That means that all the events that may occur between the sequence up-loading and its execution and may affect the user demand (arrival of urgent requests), the satellite state (information about component failures), or the environment (new meteorological forecasts), are not taken into account.

Only an on-line management, performed either on-board or on the ground via an inter-satellite communication network, can take them into account. But it is sure that such a management may induce strong temporal constraints on the reasoning task and consequently severe limitations of the search.

On-board versus on the ground management

Managing on-board has undeniable advantages: direct availability of the information about the satellite state, independence of visibility windows and of inter-satellite communication networks; complete automation and staff cost reduction ...

It has also some known drawbacks: limited computing and memory capacities available on-board, high reliability requirements on the hardware and on the software, knowing that they will run without any human supervision ...

Evaluating before deciding

Before choosing, it may be interesting to run simulations, in order to explore and to assess several trade-offs. That is what we did within a simplified framework.

A simulation framework

A simplified satellite

For the sake of rapid prototyping, we chose to consider a simplified satellite, inspired from the *Spot* family, but methods and algorithms could be extended to many more complex cases.

This satellite has only one optical instrument. We assume that it has only one degree of freedom according the roll axis for image acquisition, provided either by a mobile mirror in front of the instrument or by the satellite attitude control system. Consequently, a fixed acquisition window can be associated with each image acquisition within a given satellite revolution. We assume also that the transition time between two image acquisitions is a function of the attitude difference of the mirror or of the satellite, necessary to acquire both images. We assume finally that solar arrays and on-board recorders are such that there is neither energy nor memory constraint to check.

An off-line scheduling algorithm

We assume that the off-line computing of an optimal feasible sequence is performed before the illuminated part of each revolution for the whole of this part, from the north to the south pole, on the basis of the information available at this time. This can be easily performed by using an *Inverse Dynamic Programming* algorithm (LC78) based on the following recurrent equations:

$$\begin{array}{lll} if & R_{bef}(r) = \emptyset & \\ then & \mathbf{E}G_{opt-bef}(r) = & 0 \\ else & \mathbf{E}G_{opt-bef}(r) = & max_{r' \in R_{bef}(r)} \\ & & [\mathbf{E}G_{opt-bef}(r') \\ & & +p_e(t,r').g(r')] \end{array} \quad (1)$$

$$\begin{array}{ll} \mathbf{E}G_{opt} = & max_{r \in R} \\ & [\mathbf{E}G_{opt-bef}(r) \\ & +p_e(t,r).g(r)] \end{array} \quad (2)$$

In these equations:

- R is the current set of requests that can be performed over the whole of the illuminated part of the revolution, and $R_{bef}(r)$ is the set of requests that can be performed before performing r, taking into account transition times;
- $\mathbf{E}G_{opt}$ is the optimum expected gain that can be obtained over the whole of the illuminated part of the revolution, and $\mathbf{E}G_{opt-bef}(r)$ is the optimum expected gain that can be obtained before performing r;
- $p_e(t,r)$ is the estimated value at time t of the accomplishment probability of request r, and $g(r)$ is the gain associated with its accomplishment.

An on-line scheduling algorithm

We assume that:

- before the beginning of the illuminated part of each revolution, an optimal feasible sequence has been computed using the above algorithm;
- following any event, like either user request arrivals or removals, or new meteorological forecasts, a new feasible sequence is computed, but resources and time are no more necessarily sufficient to compute an optimal sequence over the whole of the remaining illuminated part of the revolution;
- consequently, using a kind of *anytime* approach (BD94), only an optimal feasible sequence of length h, that is involving h image acquisitions, is searched for, by beginning with $h = 1$ and incrementing h each time it is possible, that is each time no new event interrupts the current reasoning

This can be performed by using an *Inverse Dynamic Programming* algorithm based on the previous equations slightly modified:

$$\begin{array}{lll} if & [R_{bef}(r) = \emptyset] \vee [h = 0] & \\ then & \mathbf{E}G_{opt-bef}(r,h) = & 0 \\ else & \mathbf{E}G_{opt-bef}(r,h) = & max_{r' \in R_{bef}(r)} \\ & & [\mathbf{E}G_{opt-bef}(r', h-1) \\ & & +p_e(t,r').g(r')] \end{array} \quad (3)$$

$$\begin{array}{ll} \mathbf{E}G_{opt}(h) = & max_{r \in R} \\ & [\mathbf{E}G_{opt-bef}(r, h-1) \\ & +p_e(t,r).g(r) \\ & +\hat{\mathbf{E}}G_{opt-aft}(r)] \end{array} \quad (4)$$

In these equations:

- $\mathbf{E}G_{opt-bef}(r,h)$ is the optimum expected gain that can be obtained before performing r, by selecting at most h requests;
- $\hat{\mathbf{E}}G_{opt-aft}(r)$ is an estimation of the optimum expected gain that can be obtained after performing r, function of the satellite state (time, either mirror or satellite attitude ...) when ending request r's acquisition;
- $\mathbf{E}G_{opt}(h)$ is an approximation of the optimum expected gain that can be obtained over the whole of the illuminated part of the revolution, computed when searching for an optimal feasible sequence of length h.

Note that:

- everything that is computed when searching for an optimal feasible sequence of length h is reused when searching for an optimal feasible sequence of length $h+1$;
- the only benefit of an increase in the optimal sequence length is a possible better choice of the next image to acquire; the basic question in an on-line context is indeed: *What to do next ?*
- in our current implementation, everything is computed again from scratch in case of any event modifying the problem data: user request arrivals or removals, new meteorological forecasts ... but more sophisticated implementations could reuse at least one part of the previous computing;
- in the context of this very simplified satellite, the gain that results from the computing of an optimal feasible sequence of only length h is not obvious; it will be clearer in the context of more realistic satellites, where *Dynamic Programming* on large horizons will be no more practicable, off-line as well as on-line.

Experiments

All the experiments we carried out compare an off-line scheduling with an on-line scheduling, both on the illuminated part of one revolution and both performed either on the ground or on-board.

Modeling the user request flow

To carry out experiments, we need a model of the user request flow. We assume a set of N randomly generated user requests, known before the beginning of the illuminated part of the revolution and a flow of N_a added requests and N_r removed requests, randomly generated too, within the illuminated part of the revolution.

Modeling the environment

We need a model of the cloudy cover. This model takes into account the fact that indeterminism about the cloudy cover decreases with the distance t between the current time and the image acquisition time: when t is large, the accomplishment probability is approximated via weather statistics; let $p_s(r)$ be this approximation; when t decreases, the accomplishment probability evolves, either up to 1 with a probability $p_s(r)$, or down to 0 with a probability $1 - p_s(r)$.

Modeling the environment forecasts

We need a model of the meteorological forecasts. This model takes into account the fact that the distance between the accomplishment probability and its estimation by meteorological services decreases with t: the lower t, the more reliable the meteorological information. We assume that off-line and on-line schedulings use meteorological information provided respectively t_{off} and t_{on} hours before image acquisition.

Modeling the computing power

Finally, we model the difference between the computing power on the ground and on-board by a ratio $cpr_{on/off}$ between the latter and the former.

Experimental results

We carried out systematic experiments with:

- $t_{off} = 48$ and $t_{on} = 48$ (first graph), 24 (second graph) or 1 (third graph);
- $N = 150$ and $N_a = N_r = 0, 10, 20, 30, 40$, or 50;
- $cpr_{on/off} = 1/100, 1/10, 1$, or 10.

Each parameter configuration has been tested on 500 randomly generated scheduling problem instances. Each time, what has been measured is the sum of the gains associated with the user requests that have been selected, accomplished, and not removed from the current satellite order book: G_{off} for the off-line management and G_{on} for the on-line one. Each bar in the graphs of Figure 1 shows the mean gain (or loss) $(G_{on} - G_{off})/G_{off}.100$ that can be obtained by (or results from) managing the satellite on-line and eventually on-board.

We see on these graphs that:

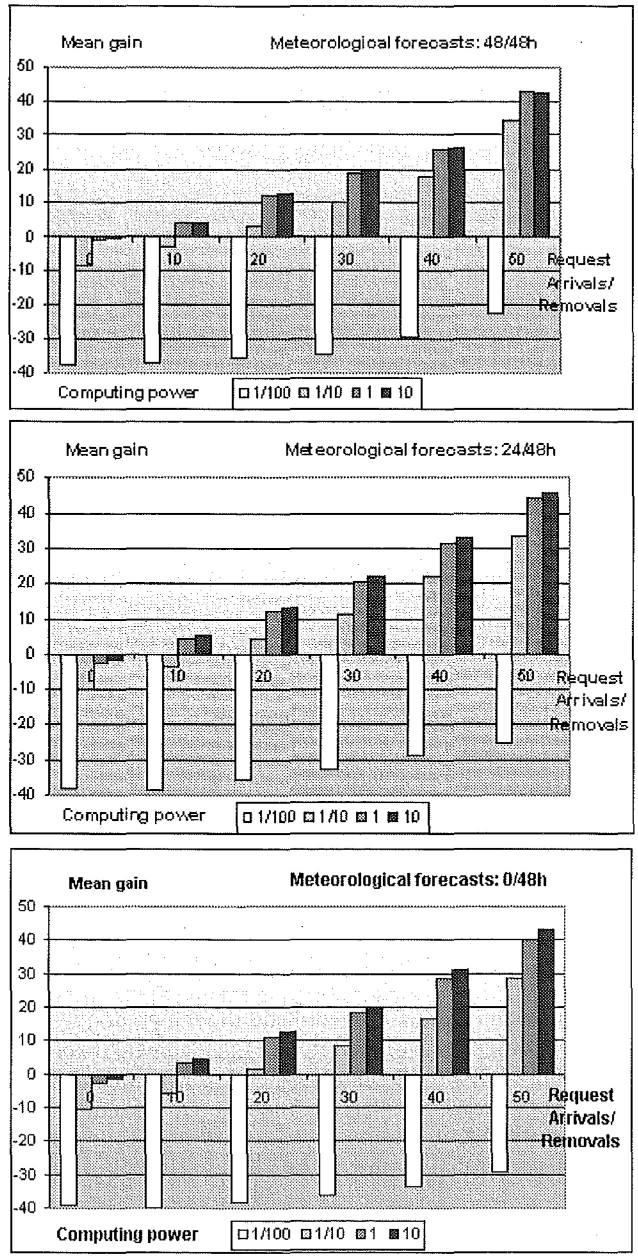

Figure 1: Mean gain (or loss) obtained by (or resulting from) managing the satellite on-line and eventually on-board: $(G_{on} - G_{off})/G_{off}.100$.

- to have at one's disposal more reliable meteorological information has no decisive influence on the results: the three graphs have globally the same shape; but the meteorological forecasting model we used may be questionable;
- the more dynamic the satellite order book, the more interesting an on-line management;
- a minimum computing power on-board is necessary to manage the satellite correctly: with a ratio $cpr_{on/off}$ equal to $1/100$ between the computing power on-board and on the ground, the gain is always negative.

Conclusion and future directions of work

The first lesson of this work is that an on-line and on-board Earth observation satellite management system is feasible and can outperform an off-line and on the ground one, in the context of a dynamic user demand (numerous requests known little time in advance), even if the computing power available on-board is lower than the one available on the ground. That means merely that, in this context as in other contexts, a limited search performed on updated data can outperform a systematic search performed on not updated ones.

The second lesson is that, at least in the context of Earth observation, building a schedule on a variable horizon, by using either *Dynamic Programming* or *Branch and Bound* methods, may be a good answer to the question of the balance between deliberation and reactivity: no frontier between them, more reactivity when it is necessary to decide quickly, more deliberation when time is available for that.

Future directions of work may concern:

- the extension of this work to more realistic models of satellite, user demand, environment, and forecasting;
- the design of more sophisticated on-line algorithms, that can, for example, reuse previous solutions or reasoning, when events slightly modify the problem data;
- the implementation on-board and the integration with the other components of the satellite management (executive, attitude and orbit control system, power management, memory management, communication management, fault diagnosis, identification and recovery ...), following the work performed for the *Proba* satellite (BDS98);
- the use of *reinforcement learning* techniques (SB98) to learn for example good approximations of the optimum expected gain that can be obtained after a given image acquisition (see section).

Acknowledgements

We would like to thank Simon Bretin, Nelly Lecubin, and Olivier Samain, from *SupAéro* (French Engineering School in Aerospace, Toulouse), who implemented the simulation framework that allows us to obtain these experimental results.

References

M. Boddy and T. Dean. Deliberation Scheduling for Problem Solving in Time-Constrained Environments. *Artificial Intelligence*, 67(2):245–285, 1994.

E. Bornschlegl, P. David, and H. Schindler. On-Board Software and Operation for Proba Small Satellite Autonomy. In *Proc. of the International Conference on Data Systems in Aerospace (DASIA-98)*, Athens, Greece, 1998.

E. Bensana, M. Lemaître, and G. Verfaillie. Earth Observation Satellite Management. *Constraints : An International Journal*, 4(3):293–299, 1999.

R. Larson and J. Casti. *Principles of Dynamic Programming*. Marcel Dekker Inc., New York and Basel, 1978.

R. Sutton and A. Barto. *Reinforcement Learning*. MIT Press, Cambridge, Massachusetts, 1998.

T. Schiex, H. Fargier, and G. Verfaillie. Valued Constraint Satisfaction Problems : Hard and Easy Problems. In *Proc. of the 14th International Joint Conference on Artificial Intelligence (IJCAI-95)*, pages 631–637, Montréal, Canada, 1995.