A Stable-Set-Based Move to Solve an Agile Earth Observation Satellite Scheduling Problem with a Local Search

Valentin Antuori, Damien T. Wojtowicz, Emmanuel Hebrard*

LAAS-CNRS, Université de Toulouse, CNRS, Toulouse, France

{vantuori, dwojtowicz, ehebrard }@laas.fr

Keywords: *Agile Earth Observation Satellite, Scheduling, Constraint-Based Local Search.*

1 Introduction and Problem Statement

Since the introduction of Pléiades [2], satellite surveillance has typically been carried out using agile Earth observation satellites (AEOS). The trend is for ever larger constellations, so that observation requests can be answered within a short delay. Therefore, the AEOS Scheduling Problem (AEOSSP) has become harder to solve, as the size of the instances grow.

A wide array of techniques have been proposed in the literature [4], leveraging exact methods, heuristics, metaheuristics and machine learning. As AEOSSP is NP-hard, most of these typically relax operational constraints (e.g. onboard energy availability), consider a smaller scope (e.g. a single satellite), or do not tackle underlying problems (e.g. download planning), so as to efficiently compute a schedule. As part of an industrial collaboration with the start-up Prométhée for its upcoming Japetus constellation, we are developing its observation scheduler.

We consider the following elements of the problem. The constellation comprises twenty AEOS, interacting with a third-party network of a dozen ground stations. Client requests associate areas of interest that should be covered during one or several time intervals (repetitions of the same request). These areas of interest are split into *meshes*. A satellite has several opportunities of acquisition within the planning horizon. An acquisition opportunity concerns one mesh, has a duration time, a time window, a certain amount of data when performed, and a profit depending on its quality (e.g. luminosity), interest (e.g. simultaneous overlapping requests), and priority. Therefore, satisfying a request implies to schedule the acquisition of the meshes covering its entire areas of interest for each of its time intervals. Multiple acquisitions of the same mesh for the same request and repetition are forbidden.

Several operational constraints must be taken into account. First, satellites have to manoeuvre between two successive acquisitions: we therefore assume that *transition time* between them is sequence-dependant. Then, they have a finite onboard memory, restricting the amount of acquisitions a satellite can perform. Memory can be freed by *downloading* the images using the downlink provided by ground stations, that can be contacted during download time windows. Each image must be downloaded as a single block, and each acquisition-download window couple has an associated profit. The overall objective is to maximize the global profit.

2 Proposed Method and Perspectives

We propose to solve the AEOSSP for Japetus with a three-phase anytime method, based on a framework inspired by the Constraint-Based Local Search approach [3]. The first phase selects a pool of requests to be satisfied, the second phase selects the acquisitions required to satisfy said requests, and the third phase ensures that memory-related constraints are respected by

*We would like to thank our industrial partner Prométhée, in particular Florian Baixas, Maxime Gratereau, Vivian Larrieu, Olivier Maes, Frédéric Tromeur and Alexandre Vauquier for their help and feedback.

computing a download planning. These phases are repeated, each iteration leveraging the last solution found at a previous iteration, gradually building a better solution by local search. A solution is valid if it fulfils all the constraints. Finally, the best solution ever found is returned.

The *request selection* phase uses the result of the previous iteration. If the latter is a feasible solution, then a request is randomly added to the pool of selected requests. If it failed, then a selected request is swapped with an unselected one.

The *acquisition selection* solves the problem of scheduling the acquisitions of the selected requests, all of which must be satisfied, with no regard to memory. As of now, transition times are not considered, and the duration of the acquisitions is assumed to be equal to their time window's. For each satellite, its sequence of acquisition opportunities can be divided into maximal-length sub-sequences between repeating acquisitions, in which we can find a locally maximal acquisition set in linear time by solving the maximum independent set problem on the associated interval graph [1]. This approach is then integrated into a local search by iteratively solving a maximum independent set problem for each sub-sequence of the satellites until a local maximum is found. Profit within each independent set problems is adjusted by granting bonuses to uncovered meshes and to acquisitions already in the current plan. This method can be extended to take into account transition times and to determine task start dates. Within the interval graphs, shorter sub-intervals could be considered so as to choose a start date with a finer grain. Arcs between all interval's sub-intervals should be added in order to prevent redundant actions to be selected. However, such a graph would not be chordal anymore, which is a necessary condition to solve the maximum independent set problem in linear time. The same algorithm can still be used for its efficiency, at the expense of its optimality guarantee.

The *memory management* phase is performed only if the previous one managed to satisfy the requests selected by the first phase. Download planning is seen, for each satellite, as a sequence of knapsack problems, one for each download opportunity, where the capacity is the download capacity, and the objects are the acquisitions still onboard. A greedy algorithm is used to solve these sub-problems, computing a download plan. If a memory overflow is detected on a satellite, another knapsack problem is solved, where the capacity is the satellite's memory, and the objects are the acquisitions onboard when the following download opportunity occurs. Non-selected acquisition are removed from the plan.

As the needs of our industrial partner are yet to be fully specified, this work remains preliminary and will be extended in order to propose an effective scheduler. There are improvement perspectives for all three phases of our method: heuristics should be defined to select requests, download planning and acquisition selection should be more integrated, and new local moves should be designed for the latter. Moreover, some meshes could conveniently be acquired together as strips so as to avoid transition times and possible geographic overlaps between meshes. Support for the strips within the acquisition selection phase is yet to be defined. Finally, onboard energy-related constraints will have to be taken into account.

References

[1] András Frank. Some polynomial algorithms for certain graphs and hypergraphs. In *Proc. of the 5th British Combinatorial Conf.*, pages 211–226, Winnipeg, 1976. Utilitas Mathematica.

[2] Michel Lemaître, Gérard Verfaillie, Frank Jouhaud, Jean-Michel Lachiver, and Nicolas Bataille. Selecting and scheduling observations of agile satellites. *Aerospace Science and Technology*, 6(5):367–381, September 2002.

[3] Laurent Michel and Pascal Van Hentenryck. Constraint-Based Local Search. In Rafael Martí, Panos M. Pardalos, and Mauricio G. C. Resende, editors, *Handbook of Heuristics*, pages 223–260. Springer International Publishing, Cham, 2018.

[4] Xinwei Wang, Guohua Wu, Lining Xing, and Witold Pedrycz. Agile Earth Observation Satellite Scheduling Over 20 Years: Formulations, Methods, and Future Directions. *IEEE Systems Journal*, 15(3):3881–3892, September 2021.