Commentary on: AIMS: A Tool for Long-term Planning of the ESA INTEGRAL Mission

Cédric Pralet and Gérard Verfaillie

by

Roman Barták
Charles University in Prague, Faculty of Mathematics and Physics, Malostranské nám. 2/25, 118 00 Praha 1, Czech Republic
bartak@ktiml.mff.cuni.cz

Paper Summary

The paper describes a tool for long-term scheduling of observations for the ESA satellite INTEGRAL. The tool generates detailed schedules for approximately one year of observations where some constraints such as visibility windows, observation patterns, and non-overlaps of observations are followed while other constraints such as resource (memory, energy) consumption and communication are ignored. The problem belongs to the area of oversubscribed scheduling where the task is to maximize the scientific output by maximizing the number of scheduled observations and their quality.

The tool has been implemented on top of a generic planning and scheduling framework called APSI (Advanced Planning and Scheduling Initiative). The schedule is constructed incrementally by adding observations to initially empty schedule. The solving method is basically a local search algorithm that combines ideas from hill-climbing (follow the greatest ascend), simulated annealing (accept even non-improving moves), and tabu search (forbid repetition of close moves). Restarts are used to diversify search.

The presented tool is currently being integrated to software for controlling scientific operations of the satellite and it is being evaluated in operational environment.

The paper starts by introducing the APSI study (Advanced Planning and Scheduling Initiative) by ESA, followed by a textual description of the problem being solved. Then a more formal description of the problem is given in terms of constraint satisfaction followed by a less formal overview of the solving method. After that the presented tool is described form the user perspective. The paper is concluded by discussion why the chosen method is appropriate for the problem.

Problem Specification

The problem from the paper is an interesting combination of detailed and aggregated scheduling. Detailed scheduling is represented by using real observations/activities that are allocated to precise times. I believe that this is due to the nature of the problem where observation windows play a crucial role. The aggregated character is hidden in disregarding some constraints such as memory allocation and communication. It would be surely interesting to discuss more this split character of the problem, namely how the decision which part of the problem must include all details and which constraints can be omitted was done. This is closely connected to the large-scale character of the problem that is due to using all observations in the horizon of one year. The second characteristic of the problem is oversubscription. I would use the word oversubscribed rather than over-constrained that is used by the authors. There are too many activities to fit in the scheduling horizon but there are also no deadlines so the activities can be postponed and no constraint is violated. Hence the focus is on the optimization part of the problem and, according to my understanding, feasibility seems easy to achieve (perhaps only the periodic observations introduce some constraints that might be harder to satisfy). These features of the problem are closely related to solving technique as I will discuss later.

Problem description is both informal and formal which is a good mixture to understand the problem. I only think that the word “observation” is a bit overused. In particular, there is some confusion between the elementary observations and the sub-operations. As I understand, the elementary operations specify discrete steps for activity duration. It might be that the terminology is given by the application domain but using similar words in slightly different meanings makes it harder to follow the text.

There is one misleading feature of the problem description – the authors interleave formal problem description with the modeling decisions. In particular, the introduction of observation activities is a decision that is outside the problem description. In particular, I mean the decision to use two observation activities per normal observation and time window. It would be useful to discuss reasons for this decision. Why using two activities and not three or more? Why using activities per time window rather than allocating a smaller number of activities to time windows? These modeling decisions are clearly important for problem solving so it is interesting to discuss them. My feeling is that the current approach might be one of the reasons why the problem is supposed to be large scale as many dummy activities are introduced. It would be interesting to see what the ratio between the number of generated observation activities and the number of used observation activities in the final schedule is. Using dummy activities also introduces some symmetries into the model, namely, for two activities in the time window, one may select either one to have one observation in that time window. Is this somehow taken in account by the solver?

From the text, it seems that the authors tested the presented approach using one data set. First, I think that it is not possible to do conclusions based on one data set unless the authors did other experiments that are not covered by the paper. From the academic perspective, it would be interesting to know whether ESA is willing to make the data set(s) publicly available to researchers outside the APSI consortium. I think that public availability of benchmark data sets would contribute both to industry (ESA), that can obtain access to broader research results, and to academia, that can obtain real-life challenges of practical interest. Regarding the data set described in the paper, in addition to its size, it would be also interesting to see the distribution of different observation types as they differ in constraints (for example, periodic observations seem to impose harder constraints).

Solving Approach

While, I was generally satisfied with the problem description, I have found the description of the solving techniques a bit vague. The general principles are clear, but technical details are missing which makes it hard to understand how it works. Hence, I have more questions than general comments regarding the solving technique. First, it is not clear what local moves are assumed. Clearly, it is possible both to add and to remove activities from the schedule in the style of Iterative Forward Search by Tomáš Müller, but it is not clear how the move is realized. How the activity to be removed is selected? Similarly, how the activity to be added is selected? These questions are also related to the question how the local moves are evaluated. Are you using the overall utility function to evaluate each move? Then clearly, adding activities is preferred as it increases utilization of resources. As the number of all possible local moves is large, the authors suggest using only a small subset of them, but how is this subset selected? Tabu search is used to avoid local cycles, but what is the information stored in the tabu list? It seems than one local move influences more activities so it is not clear what information is exactly stored in the tabu list. The authors suggest using restarts to diversify search. However, using restarts assumes some stochastic element so the search algorithm does not follow exactly the same trace after the restart. This is not mentioned in the paper so it is not clear how it works.

There is one important aspect of the work (from the perspective of research collaboration) and this is exploiting the common general APSI framework developed by other partner. My feeling is that the connection between the presented techniques and APSI framework is rather loose. As I understand the APSI framework is used merely to check overlaps of time intervals which can be done easily without any specific framework. It would be interesting to see more details about what services are provided by the APSI framework and how these services are exploited by the AMSI tool.

At the beginning of the paper the authors promised to describe some lessons learned from the project. There is indeed a discussion on using alternative approaches such as CP Optimizer and COMET, but this is too narrative. I would like to see more rigorous comparison of the techniques. For example, my feeling is that the problem is not very constrained so constraint propagation cannot infer a lot of pruning and hence constraint programming might not be appropriate. This might be a stronger reason than the reason of being a large-scale problem mentioned by the authors. Let me comment on it a bit. The graph in Figure 5 suggests a fast increase of solution quality as the observation actions are being added to the schedule. This increase goes up to 0.95 which is not that far from the best obtained result (0.98). It is surprising the CP cannot use exactly the same strategy when instantiating the variables that corresponds to allocation of activities to the schedule. The authors report quality obtained by the CP Optimizer to be at most 0.90. Isn't it because it is not possible to influence the search strategy of the CP Optimizer? Or am I wrong when I suppose that the initial increase of solution quality is realized by adding new observations (at least, this is what the authors also wrote)? I am also a bit confused by the explanation why COMET cannot be used – missing scheduling constructs. When looking at the constraint model in the paper, there are no particular scheduling constructs mentioned so isn't it possible to encode these constraints in COMET? I think that it would be useful to see which particular constructs are hard for COMET as this development environment seems to me very appropriate to implement the solving approach presented in the paper.

Application Perspective

From the application perspective, it would be interesting to know how the presented tool is supposed to be integrated into existing infrastructure. It looks like it is a standalone tool, but I suppose that just its optimization core (without the presented GUI) will be used in ESA. How the data will be communicated between the tool and the rest of the system? Is the communication realized via files or is a direct access to database used? In general it would be interesting to know what the current practice to integrate optimization components to existing software is.

I was a bit surprised that it is not easy to add new constraints to the tool as it requires modifying the solving algorithm. Does it complicate exploitation of the tool? My experience is that customers rarely specify all constraints in advance and there is always some "one more thing". Does the customer prefer a "one-shot" type of optimization software or a more generic one that can be applied after some customization to a wider range of problems?

The results are very interesting in terms of quality. The open question is whether this quality corresponds to what is expected and whether the long-term schedule can be realized after adding the omitted constraints.