Data-driven Onboard Scheduling for an Autonomous Observation Satellite

Chao Li[1], **Patrick De Causmaecker**[1], **Yingwu Chen**[2]
[1] CODeS, Department of Computer Science, KU Leuven
[2] College of System Engineering, National University of Defense Technology
{chao.li, patrick.decausmaecker}@kuleuven.be, ywchen@nudt.edu.cn

Abstract

Observation requests for autonomous observation satellites are dynamically generated. Considering the limited computing resources, a data-driven onboard scheduling method combining AI techniques and polynomial-time heuristics is proposed in this work. To construct observation schedules, a framework with offline learning and onboard scheduling is adopted. A neural network is trained offline in ground stations to assign the scheduling priority to observation requests in the onboard scheduling, based on the optimized historical schedules obtained by genetic algorithms which are computationally demanding to run onboard. The computational simulations show that the performance of the scheduling heuristic is enhanced using the data-driven framework.

1 Introduction

To quickly respond to dynamic observation requests and other unexpected events, autonomy is vital for observation satellites orbiting the Earth with high-resolution optimal instruments. For this reason, autonomous satellites have been studied and tested as agents which are capable of making decisions onboard and operating without help from ground stations [Chien *et al.*, 2005].

The challenge to achieve a high level of autonomy is the onboard scheduling based on limited computational resources. To handle the observation requests which are dynamically generated from the autonomous detection of new targets to be further observed [Chien and Troesch, 2015], frequent rescheduling is required to resolve timing or storage conflicts. And the optimization objective of the onboard scheduling is to maximize the total profits of observation requests completed.

The offline scheduling of an agile observation satellite is NP-hard [Lemaître *et al.*, 2002]. The onboard scheduling is more complicated, given that the set of observation requests is not static. In the onboard scheduling settings, heuristics with low complexity are more widely used than exact algorithms that find optimal solutions at a high computational cost [Chu *et al.*, 2017].

In this work, AI techniques are utilized to improve the onboard scheduling heuristic which is based on the earliest possible start time of observation requests, by exploiting past scheduling data and optimized schedules.

2 The Data-driven Onboard Scheduling

In this section, we present a data-driven onboard scheduling method which is fast using limited computing resources. As illustrated in Figure 1, the procedure of the proposed method consists of an offline learning phase and an onboard scheduling phase.

The offline learning in the ground stations is key to improving the optimization ability of the onboard scheduling heuristic. Optimization methods such as genetic algorithms are applied to generate optimized schedules based on past scheduling data. Then a classifier, called the request filter, is trained to learn patterns of the observation requests in the optimized schedules.

In the onboard scheduling, the request filter trained offline is used to assign scheduling priorities to the new and existing observation requests according to the features of observation requests and satellite's status. The requests with higher priority will be scheduled first. The scheduling heuristic we employed is based on a greedy method proposed in [Chien and Troesch, 2015] with $O(n^2)$ complexity, which builds a schedule according to the earliest possible start time of observation requests.

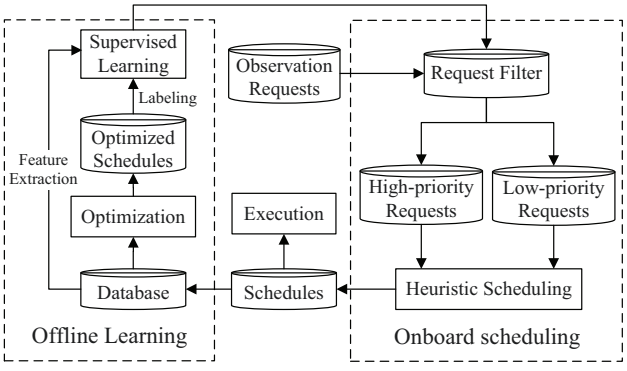

Figure 1: Procedure of the data-driven method

Algorithm 1 Offline Learning

Intput: the past requests set R_0 and satellite state S_0
$OptimizedSchedule \leftarrow GeneticAlgorithm(R_0, S_0)$
for request r in R_0 **do**
  if r *is in* $ImprovedSchedule$ **then** //Check whether request r is scheduled successfully by GA
    $Label(r) \leftarrow 1$
  else
    $Label(r) \leftarrow -1$
  end if
end for
$TrainSet \leftarrow FeatureExtraction(R_0, S_0, Label)$ //Construct the training set
$RequestFilter \leftarrow TrainNeuralNet(TrainSet)$ //Train a Neural Network classifier
return $RequestFilter$

Algorithm 1 details the steps of offline learning. Firstly, the original schedules generated onboard are optimized by a genetic algorithm using the historical scheduling information. It is noted that the genetic algorithm can be replaced with other optimization methods. To construct a training set from the optimized schedules for the request filter, every request is labeled as positive or negative, depending on whether it is arranged successfully in the optimized schedules. The extracted features include the attributes of observation requests (observation profit, required observation time, executable time intervals, the maximum observation angle and the conflict degree) and the status of the satellite (observation angle, storage and energy) at the moment of scheduling. The conflict degree of a request is a binary value to indicate whether its executable time intervals overlap other requests' executable time intervals. Finally, the request filter is learned as a back-propagation (BP) neural network classifier, using the training set generated above.

The reason why the BP neural network model is applied as the learning algorithm is that it obtained the best test accuracy in our configuration experiments, compared with other models such as decision trees and the SVM. The onboard prediction of a neural network after training is just a sum of products. Therefore, using the request filter trained offline, the onboard scheduler can assign priorities to observation requests for the heuristic onboard scheduling efficiently, with the objective of maximizing the total observation profit of completed requests.

3 Simulations

The simulations were conducted in the MATLAB. Two onboard scheduling methods for an agile observation satellite were tested, including the proposed data-driven onboard scheduling method (DDOS) and the greedy heuristic scheduling method (GHS) which arranges each observation request as early as possible without using the data-driven framework. Dynamic observation requests were generated randomly within a simulation period of 45 minutes, and the average performance over six runs was recorded.

The simulation results in Table 1 validate that DDOS and GHS are both fast. However, with the increase of the number of dynamic observation requests, there is a significant difference in their optimization abilities in terms of the total profit of schedules and the number of observation requests completed. The total profit of DDOS is at least 23% more than that of GHS, in the scenarios with 50 and 70 requests where the observation requests cannot be completed totally.

Number of requests	Scheduler	Requests completed	Total profit	Running time (seconds)
30	DDOS	30.00	598.67	0.35
	GHS	30.00	598.67	0.04
50	DDOS	38.33	866.17	0.71
	GHS	34.67	701.67	0.06
70	DDOS	48.50	1185.83	0.12
	GHS	39.67	908.17	0.08

Table 1: The simulation results

4 Conclusions and Future Work

This research studied the AI techniques to be used in the onboard scheduling for an observation satellite. An offline learning classification model for assigning scheduling priorities to observation requests was incorporated in a data-driven framework, using the information concerning past observation requests and optimized schedules from genetic algorithms. The experiments validated the effectiveness of the proposed data-driven method for the onboard scheduling.

In the future, more heuristics such as local search will be introduced into the data-driven framework to enhance the optimization capability. We also plan to study the features for training a more accurate classification model. Another direction is to apply the proposed method to other applications, for example, the multi-satellite scheduling and other machine scheduling problems.

References

[Chien and Troesch, 2015] Steve Chien and Martina Troesch. Heuristic onboard pointing re-scheduling for an earth observing spacecraft. In *Proceedings of the 25th International Conference on Automated Planning and Scheduling (ICAPS-15)*, 2015.

[Chien *et al.*, 2005] Steve Chien, Rob Sherwood, Daniel Tran, Benjamin Cichy, Gregg Rabideau, Rebecca Castano, Ashley Davis, Dan Mandl, Bruce Trout, Seth Shulman, et al. Using autonomy flight software to improve science return on earth observing one. *Journal of Aerospace Computing, Information, and Communication*, 2(4):196–216, 2005.

[Chu *et al.*, 2017] Xiaogeng Chu, Yuning Chen, and Yuejin Tan. An anytime branch and bound algorithm for agile earth observation satellite onboard scheduling. *Advances in Space Research*, 60(9):2077–2090, 2017.

[Lemaître *et al.*, 2002] 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, 2002.