Data-Driven Approach for Satellite Onboard Observation Task Planning Based on Ensemble Learning

Shuang Peng, Jiangjiang Wu, Chun Du, Hao Chen(✉), and Jun Li

College of Electronic Science and Technology, National University of Defense Technology, Changsha, Hunan, China
{pengshuang08,hchen}@nudt.edu.cn

Abstract. Onboard task planning can enhance the responsiveness of satellite to dynamic changes, which has attracted widespread attention. In this paper, the Satellite Onboard Observation Task Planning (SOOTP) problem is studied, and a data-driven onboard planning approach is proposed to decide the observation task to execute in real-time using machine learning techniques. In the approach, the satellite can learn how to make optimal decisions from the historical planning results. What is more, we design five types of features and employ three ensemble learning algorithms to solve the SOOTP. A comparison of the proposed method against two online searching algorithms indicates that the former has smaller profit gap and shorter response time, which verify the feasibility of our method.

Keywords: Satellite autonomy · Observation Task Planning · Data-driven onboard planning · Machine learning · Ensemble learning

1 Introduction

Onboard task planning has received extensive attention in recent years due to its rapid response capability to emergencies and dynamic changes (such as the arrival of new observation tasks [1], the unexpected resource level [2]), and it has become a research hotspot in the field of satellite task planning.

As shown in Fig. 1, the Autonomous Earth Observation Satellite (AEOS) orbits the Earth. It needs to decides whether to observe ground targets when passing over them (**observation** task) and download data when it passes over the ground station (**transmission** task). Energy is consumed when executing the observation tasks and transmission tasks, and is obtained only when the AEOS is in the sun. Memory is consumed when performing the observation tasks and is freed when performing the transmission tasks. In this paper, we assuming the used transmission tasks have already been determined from the ground in advance, so we only considering the Satellite Onboard Observation Task Planning (SOOTP) problem.

Q. Wu et al. (Eds.): WiSATS 2020, LNICST 358, pp. 175–184, 2021.
https://doi.org/10.1007/978-3-030-69072-4_15

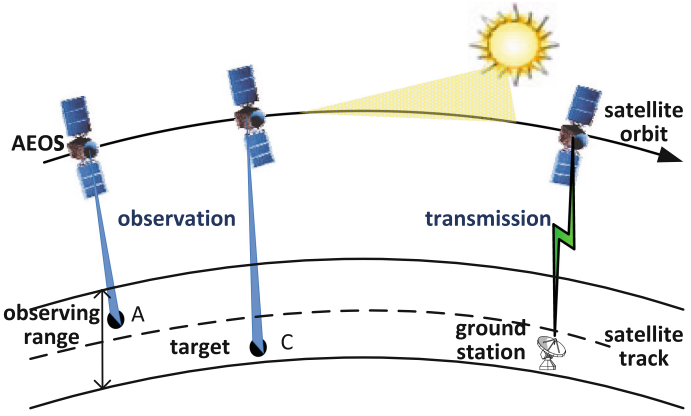


Fig. 1. Illustration of the satellite onboard tasks planning problem

The majority of existing studies solve the SOOTP problem with searching algorithms based on the rolling/continuous planning architecture [1]. In this architecture, the onboard planner only schedules the observation tasks in a short-term rolling planning horizon. As time goes on, the resource and tasks in the rolling planning horizon changes, it will trigger the scheduling process and repair the plan dynamically. For example, Chien et al. [2,3] considered the changes of observation tasks and unexpected resource level, and proposed an iterative repair algorithm. Beaumet et al. [4] used heuristic rules and iterated stochastic search algorithm to solve the problem. Li et al. [5] designed two heuristic rules of profit first and resource utilization first. The above researches mainly use heuristic methods to repair the plan in the rolling planning horizon. However, these repair strategies are relatively simple. There is still room for improvement in the optimization of plan.

With the rapid development of machine learning technologies, researchers try to combine machine learning methods with search algorithms to solve the SOOTP problem. They use machine learning methods to learn empirical knowledge from historical planning data to improve the performance of search algorithms. For example, Li et al. [6] used the neural network to compute the scheduling priority of user's requests, and combined it with the heuristic algorithm. Lu et al. [7] used a classifier to compute the probability of a task can be arranged and combined it with the greedy algorithm. The experimental results of the above studies indicate that the learning-based search algorithm is superior to the rule-based heuristic search algorithms. However, the optimization of these algorithms is limited by the architecture of the search algorithms, and the potential of the machine learning methods is not fully utilized.

In this paper, a new data-driven approach is proposed. It can use the machine learning method to solve the SOOTP problem without relying on any searching algorithms. The contribution of this study includes: 1) we propose a data-driven onboard planning framework, which can decide whether an observation task

should be scheduled only through machine learning techniques; 2) we design five types of features for classification and use three ensemble learning methods to learn how to make a such decision; 3) Experiments results demonstrate that our approach can achieve smaller profit gap and shorter response time.

2 Problem Formulation

The objective of our problem is to maximize the total profit of the scheduled observation tasks without violating the constraints. The following is the notations we will use.

- $[Ta, Te]$: the planning horizon.
- M: the set of observation tasks.
- O: the set of orbits.
- M_o: the set of observation tasks in orbit $o, o \in O$. $\sum_{o \in O} M_o = M$ and $M_o \cap M_p = \varnothing (o, p \in O, o \neq p)$
- ta_i: the start time of task $m_i, m_i \in M$.
- te_i: the end time of task $m_i, m_i \in M$.
- p_i: the profit of task $m_i, m_i \in M$.
- e_i: the energy required to perform task $m_i, m_i \in M$.
- d_i: the memory required to perform task $m_i, m_i \in M$.
- Ts_{ij}: the setup time between tasks m_i and task m_j.
- $e(t), d(t)$: the energy level and data level at time t. The computation process can refer to Reference [8,9].
- E_{min}, E_{max}: the lower bound and upper bound of energy level.
- D_{min}, D_{max}: the lower bound and upper bound of memory level.
- $x_i \in \{0, 1\}$: the decision variable. $x_i = 1$ indicates task m_i is selected, otherwise not.

The mathematical model is as follows.

$$\text{Maximize} \quad \sum_{m_i \in M} x_i * b_i \tag{1}$$

$$\text{Subject to} \quad E_{min} \leq e(t) \leq E_{max} \tag{2}$$

$$D_{min} \leq d(t) \leq D_{max} \tag{3}$$

$$ta_j - te_i \geq Ts_{ij} * x_i * x_j \tag{4}$$

The objective (1) is to maximize the total profit. Constraint (2) restrict the scope of the energy level. Constraint (3) restrict the scope of the memory level. Constraint (4) requires the time interval between any two scheduled tasks should be greater than the setup time.

3 Data-Driven Onboard Planning Framework

To improve the responsiveness of AEOS, we propose a data-driven onboard planning framework, which using the machine learning techniques to decide whether an observation task should be performed. As illustrated in Fig. 2, the framework consists of offline learning and online decision-making.

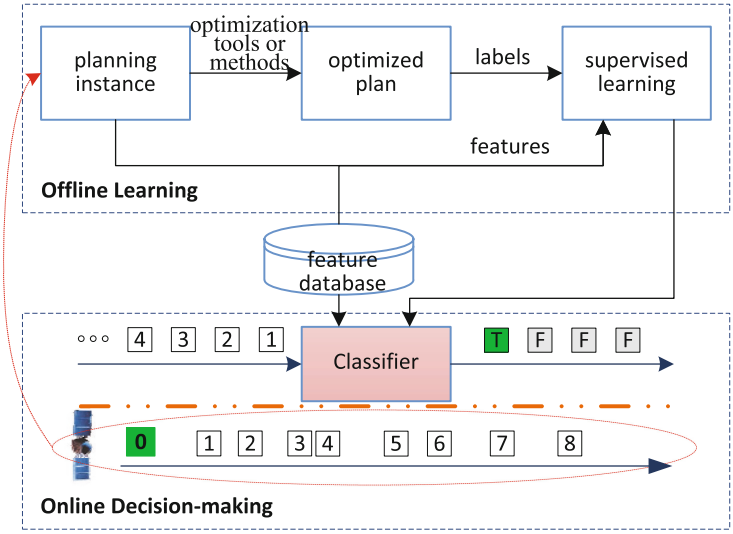


Fig. 2. Data-driven onboard planning framework

In the offline learning part, the onboard planning instance is optimized by the optimization tools or methods such as CPLEX solver. The label of an observation task is '1' if it is in the optimized plan, otherwise is '0'. Then we train the decision-maker (e.g., a classifier) to learn the decision rules for judging whether an observation task can be arranged. Details of offline learning can refer to Algorithm 1. Considering the limited computing power of the onboard computer, this part can be done on the ground.

In the online decision-making part, the decision-maker is employed to make decisions on the observation tasks one by one in chronological order. An observation task is selected if the output is 'true', otherwise it is discarded.

To train the decision-maker, the main problem we face is that how to construct a fixed-size length features to represent the observation task to be decided. Thus, we design five class of features based on the empirical knowledge and use them for classification. Before introducing these features, we first give the definition of group feature vector.

Definition 1. ***Group Feature Vector (GFV).*** *Let M_s be the subset of M, then the feature vector of M_s is represented by a six-tuple $GFV(M_s) = (n, dc, ad, ap, ae, am)$:*

Algorithm 1. Offline learning

Require: planning instance M, optimized plan P.
Ensure: classifier
1: $samples \leftarrow null$
2: **for all** task $m \in M$ **do**
3: **if** $m \in P$ **then**
4: $label(m) \leftarrow 1$
5: **else**
6: $label(m) \leftarrow 0$
7: **end if**
8: $feature(m) \leftarrow computeFeatures(m)$
9: $samples \leftarrow samples \cup (feature(m), label(m))$
10: **end for**
11: $classifier \leftarrow Training(samples)$

- $n = |M_s|$: *the number of observation tasks in* M_s.
- $dc = \sum_{m_k \in M_s} \sum_{m_l \in M_s} b_{kl}/n^2$: *the degree of conflict between tasks.* ($b_{kl} = 1$ *if tasks* m_k *and* m_l *violate the constraint* (4)*, otherwise is 0)*
- $at = \sum_{m_k \in M_s} (te_k - ta_k)/n$: *the average duration of observation tasks in* M_s.
- $ap = \sum_{m_k \in M_s} p_k/n$: *the average profit of observation tasks* M_s.
- $ae = \sum_{m_k \in M_s} e_k/n$: *the average energy required to perform observation tasks* M_s.
- $ad = \sum_{m_k \in M_s} d_k/n$: *the average memory required to perform observation tasks* M_s.

Let $m_i \in M_o$ be the observation task to be decided, then the following five group of features (i.e., totally 36 features) are computed.

1. state features $(e(ta_i), d(ta_i))$: the energy level and data level at the start time of m_i.
2. task features $(p_i, te_i - ta_i, e_i, d_i)$: the description of m_i.
3. local features $GFV(M_c)$: the GFV of task set M_c, where M_c is the set of observation tasks that conflict with m_i. It represents the degree of conflict of m_i with other observation tasks.
4. global features $GFV(M_o), \ldots, GFV(M_{o+k-1})$: the GFV of task sets $M_o, \ldots, M_{o+k-1}$. It is used to represent the distribution of the observation tasks in subsequent orbits, which can avoid the onboard planner making short-sighted decisions. k is set to 3 in this paper.
5. transmission features $GFV(M_t)$: the GFV of task set M_t, where M_t is the set of observation tasks before the next transmission task. It is used to represent the degree of competition between m_i and other observation tasks on memory resource.

Finally, we use three ensemble learning algorithms, including Extreme Gradient Boosting (XGBoost) [11], Gradient Boosting Decision Tree (GBDT) and Random Forest (RF) [12], to train the decision-maker in the framework.

Table 1. Data set

Scenarios	s100	s200	s300	s400	s500
Number of instance	120	120	120	120	120
Number of tasks in a instance	100	200	300	400	500

4 Experiment

4.1 Experimental Setting

In our experiment, the planning horizon was set as 24 h, and five scenarios with totally 600 planning instances are randomly generated. Details were shown in Table 1.

All experiments were run on a Windows 7 OS (Intel i7-4712M, 2.30 GHz, 8 GB RAM). All algorithms were implemented in Python except CPLEX. The decision-maker is implemented in Python's package such as scikit-learn [14] and xgboost [15].

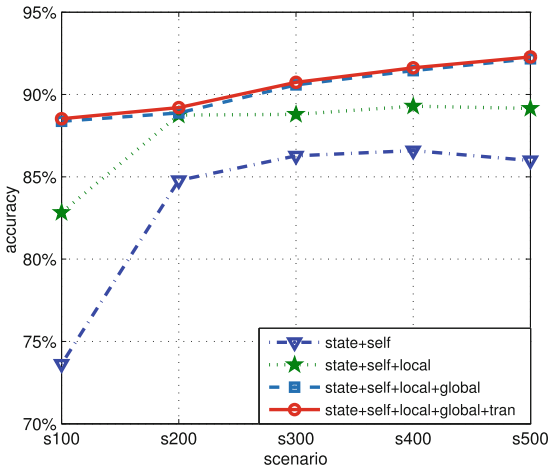
To measure the performance of these algorithms, four statistics [10], include accuracy, response time, average profit and profit gap, were used.

4.2 Importance of Features

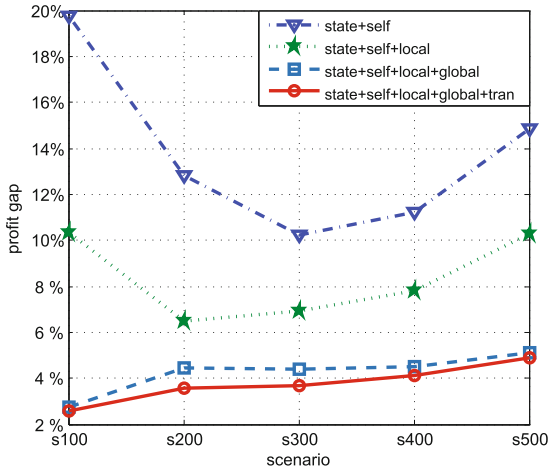
These three ensemble learning algorithms can calculate the importance of each feature while training the classifier. We use GBDT as the representative to test the impact of these feature on the performance of our method.

We first tested the effects of these five types of features on accuracy and profit gap (Fig. 3). The horizontal axis represents the planning scenarios and the vertical axis represents the accuracy or profit gap. In the legend, 'state', 'self', 'local', 'global', 'tran' represent the state features, task features, local features, global features, transmission features, respectively. In Fig. 3, we see that the accuracy increases with the use of more types of features, which proves the rationality of these five group of features. It is worth noting that the use of local features and global features results in a significant decrease in the profit gap. The local features and global features enable the AEOS to make more optimal decisions. To further improve the performance of this method, more attention can be paid to the design of these two types of features in the future. The remaining feature type combinations also show the same trend except the four feature type combinations listed in Fig. 3, and the results are no longer displayed here.

Next we analyzed the impact of each feature on the profit gap based on their importance. We sorted these features by their importance from large to small, then added each of them to a subset in order and tested the impact of the features in the subset on profit gap. The profit gap of GBDT under different number of features is shown in Fig. 4. In Fig. 4, the profit gap decreases as more features are selected. The profit gap is no longer is significantly reduced



(a) Accuracy



(b) Profit gap

Fig. 3. Accuracy and profit gap of GBDT under different type of features

when the number of features in the subset exceeds 16. It indicates that only part of features have an important impact on the performance of our method. Finally, we choose the top 16 features for our framework. In feature engineering, different combinations of features can achieve different results, and we will do more analysis on the combination of features in the future.

4.3 Comparison

To verify the effectiveness of our approach, CPLEX [13] was used for baseline and two online planning algorithms: the Profit-based First (PF) heuristic algorithm

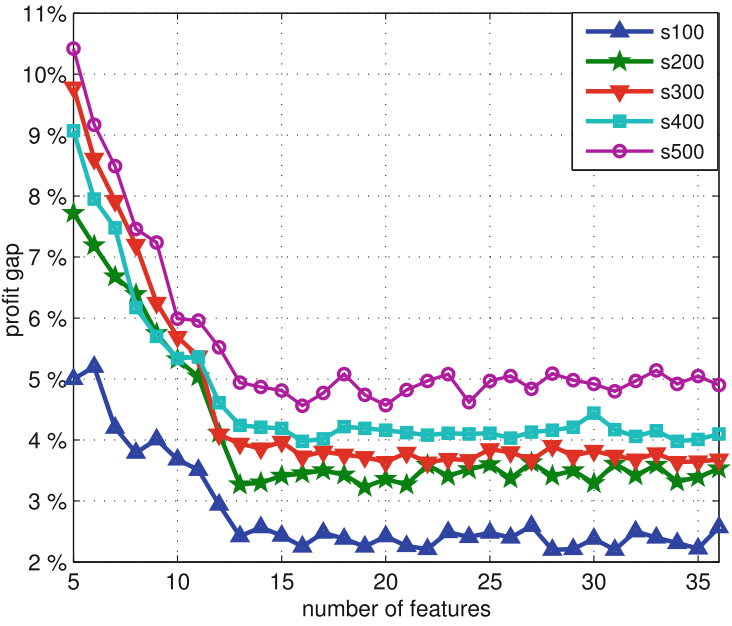


Fig. 4. Profit gap of GBDT using different number of features

and the Dynamic Profit System Benefit (d-PSB) heuristic algorithm [1] were used for comparison. The PF and d-PSB give priority to the observation tasks with higher profit and resource utilization from the task set, respectively. For more details about these two searching algorithms, refer to Reference [1].

Table 2 shows the average profit, profit gap and average time of six algorithms. In Table 2, the profit gap of RF (3.2%–6.5%), GBDT (2.3%–4.6%) and XGBoost (2.7%–4.6%) are less than that of PF (1.7%–25.8%) and d-PSB (5.8%–9.5%). It is because three ensemble learning methods have learned useful decision knowledge from historical planning data. The decision model learned by the machine learning method contains a lot of decision knowledge that cannot be expressed by rules and it is more comprehensive than the rules defined by humans. In contrast, the heuristic rules of d-PSB and PF are much simpler so that they are difficult to obtain optimal solutions. In addition, the response time of RF, GBDT and XGBoost are on the level of 10^{-4} or 10^{-3} s. Thus, RF, GBDT and XGBoost can quickly generate an feasible plan even if the computing power drops to one thousandth [1,5], which investigate the feasibility of the proposed method.

In these three ensemble learning methods, the profit gap and response time of GBDT and XGBoost is far less than that of RF, which indicate the boosting method is more suitable for our problem.

Table 2. Average profit, profit gap and response time of CPLEX, PF, d-PSB, RF, GBDT and XGBoost

Scenarios	CPLEX	PF	d-PSB	RF	GBDT	XGBoost
s100	1534	**1508 (1.7%)**	1445 (5.8%)	1486 (3.2%)	1500 (2.3%)	1493 (2.7%)
	3.4 s	0.4 ms	0.4 ms	5.5 ms	0.3 ms	0.3 ms
s200	2190	1976 (9.8%)	2011 (8.2%)	2070 (5.5%)	**2115 (3.5%)**	2113 (3.5%)
	42 s	0.5 ms	0.6 ms	4.8 ms	0.3 ms	0.3 ms
s300	2578	2148 (16.7%)	2359 (8.5%)	2436 (5.5%)	2482 (3.7%)	**2485 (3.6%)**
	174 s	0.5 ms	0.7 ms	4.6 ms	0.4 ms	0.4 ms
s400	2849	2235 (21.6%)	2583 (9.3%)	2697 (5.3%)	**2736 (4.0%)**	2732 (4.1%)
	438 s	0.5 ms	0.7 ms	4.3 ms	0.4 ms	0.4 ms
s500	3069	2277 (25.8%)	2778 (9.5%)	2870 (6.5%)	**2930 (4.6%)**	2927 (4.6%)
	1292 s	0.5 ms	0.8 ms	4.1 ms	0.5 ms	0.5 ms

5 Conclusion

In this paper, a data-driven onboard planning framework combining ensembles learning technology is proposed. It can solve the onboard observation task planning problem only through machine learning methods without rely on any searching algorithms. The experiments demonstrated that the proposed approach can achieve smaller profit gap and shorter response time.

In the future, we plan to use deep neural networks to automatically extract features and improve the performance of the decision-making model.

Acknowledgements. This work is supported by the Natural Science Foundation of Hunan Province under Grant 2020JJ4103 and National Natural Science Foundation of China under Grant 61806211.

References

1. Chu, X., Chen, Y., Tan, Y.: An anytime branch and bound algorithm for agile earth observation satellite onboard scheduling. Adv. Space Res. **60**(9), 2077–2090 (2017)
2. Chien, S., Knight, R., Stechert, A., Sherwood, R., Rabideau, G.: Using iterative repair to improve the responsiveness of planning and scheduling. In: International Conference on Artificial Intelligence Planning Systems, pp. 300–307 (2000)
3. Chien, S., Doubleday, J., Thompson, D.R.: Onboard mission planning on the intelligent payload experiment (IPEX) cubesat mission. In: International Workshop on Planning and Scheduling for Space (2013)
4. Beaumet, G., Verfaillie, G., Charmeau, M.C.: Feasibility of autonomous decision making on board an agile earth-observing satellite. Comput. Intell. **27**(1), 123–139 (2011)
5. Li, G., Xing, L., Chen, Y.: A hybrid online scheduling mechanism with revision and progressive techniques for autonomous earth observation satellite. Acta Astronaut. **140**, 308–321 (2017)

6. Li, C., Chen, Y., Causmaecker, P.D.: Data-driven onboard scheduling for an autonomous observation satellite. In: The 27th International Joint Conference on Artificial Intelligence, pp. 5773–5774 (2018)
7. Lu, J., Chen, Y., He, R.: A learning-based approach for agile satellite onboard scheduling. IEEE Access **8**(99), 16941–16952 (2020)
8. Spangelo, S., Cutler, J., Gilson, K., Cohn, A.: Optimization-based scheduling for the single-satellite, multi-ground station communication problem. Comput. Oper. Res. **57**, 1–16 (2015)
9. Peng, S., Chen, H., Li, J., Jing, N.: Approximate path searching method for single-satellite observation and transmission task planning problem. Math. Probl. Eng. **3**, 1–16 (2017)
10. Peng, S., Chen, H., Li, J., Jing, N.: Onboard observation task planning for an autonomous earth observation satellite using long short-term memory. IEEE Access **6**, 65118–65129 (2018)
11. Chen, T., Guestrin, C.: XGBoost: a scalable tree boosting system. In: The 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2016, pp. 785–794. ACM (2016)
12. Zhou, Z.: Ensemble Methods: Foundations and Algorithms. Taylor & Francis, Milton Park (2012)
13. CPLEX: IBM ILOG CPLEX optimization studio (2012). http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/index.html
14. https://scikit-learn.org/stable/
15. https://github.com/dmlc/xgboost