LETTER

Autonomous Agile Earth Observation Satellite Mission Planning with Task Clustering

Xiaohe HE[†,††,†††a)], ***Student Member***, **Zongwang LI**[†,††], **Wei HUANG**[†,††,†††], **Junyan XIANG**[†,††,†††], **Chengxi ZHANG**[††††], **Zhuochen XIE**[†,††b)], ***and*** **Xuwen LIANG**[†,††,†††c)], ***Nonmembers***

SUMMARY Agile Earth observation satellite (AEOS) mission planning (AEOSMP) problem aims to optimize observation efficiency by selecting and scheduling tasks from the Earth's surface, subject to complex resource constraints. Increased flexibility of AEOS presents challenges for autonomous mission planning and scheduling. Deep reinforcement learning (DRL) and clustering tasks are two approaches to enhance the autonomy and observation efficiency of AEOSMP. This letter introduces two innovative algorithms to tackle the AEOSMP problem: the Sequential Clique Clustering and PPO Planning algorithm (SCC-PPO) and the Simultaneous Clustering and Planning PPO Algorithm (SCP-PPO). SCC-PPO initially partitions the mission tasks into cliques, followed by planning. In contrast, SCP-PPO combines clustering and planning into a single, concurrent process. Numerical simulations reveal that SCP-PPO enhances the observation reward by 1.01% to 11.43% compared to SCC-PPO.

key words: *earth observation, agile satellite, mission planning, task scheduling, task merging, deep reinforcement learning*

1. Introduction

1.1 Backgrounds

Earth observation satellites (EOS) play a crucial role in global monitoring and management, providing essential data for tracking environmental changes, predicting weather patterns, managing disasters, and optimizing agricultural practices. These satellites are integral to various applications, including natural resource management, urban planning, environmental protection, and scientific research advancement. AEOS, in particular, represents a significant technological leap forward. Unlike their traditional counterparts, AEOS can execute rapid and frequent attitude adjustments across multiple Euler axes, substantially enhancing their operational flexibility and observation capabilities.

As the demand for high-resolution, timely observational data escalates and satellite constellations expand, the development and implementation of autonomous, efficient mission planning methods for agile satellite constellations have become increasingly imperative [1]. This growing need has led to intensified research efforts in satellite mission planning over the past two decades.

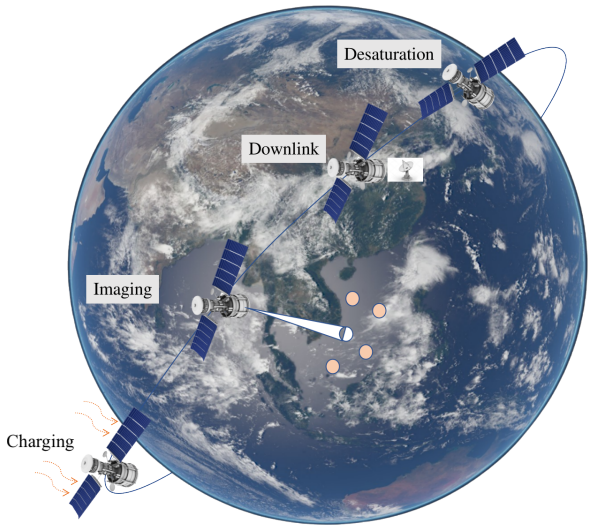

Fig. 1 Description of AEOSMP problem. Agile satellite chooses actions from charging, imaging, downlink, and desaturation modes.

Since its initial formulation by Lemaitre [2], the satellite mission planning problem has been the subject of extensive research. Scholars have proposed various solution methods, encompassing exact algorithms, heuristic approaches, and meta-heuristic techniques. However, both exact and heuristic methods exhibit inherent limitations in dynamic environments due to their open-loop nature, which precludes real-time adaptation to changing conditions. Consequently, when an observation plan fails, or new observation requirements emerge, these methods necessitate computationally expensive re-computation of a new plan.

Recent works [3]–[5] have demonstrated the significant potential of DRL in the AEOSMP problem and offered adaptive and robust solutions. Additionally, some researchers have proposed clustering methods for the AEOSMP problem to improve observation efficiency [6].

This letter presents two novel algorithms to address the AEOSMP problem: SCC-PPO and SCP-PPO. The SCC-PPO algorithm divides mission tasks into cliques and then plans the sequence. On the other hand, the SCP-PPO algorithm integrates clustering and planning into a unified,

Manuscript received November 29, 2024.
Manuscript publicized December 16, 2024.
[†]University of Chinese Academy of Sciences, No.1 Yanqihu East Rd, Huairou District, Beijing, 101408, China.
[††]Innovation Academy for Microsatellites of CAS, No.1 Xueyang Rd, Pudong District, Shanghai, 201304, China.
[†††]Shanghaitech University, 393 Middle Huaxia Rd, Pudong District, Shanghai, 201210, China.
[††††]Jiangnan University, Lihu Avenue, Wuxi, 214122, China.
a) E-mail: hithxh@gmail.com
b) E-mail: xiezc@microsate.com (Corresponding author)
c) E-mail: liangxw@shanghaitech.edu.cn (Corresponding author)
DOI: 10.1587/transfun.2024EAL2106

simultaneous process. This integration enables satellites to dynamically adjust to evolving missions and conditions, thereby enhancing observation efficiency, particularly in scenarios with high target density.

2. Problem Statement

First, the problem objectives, constraints, and underlying simulation models are described in detail. Then, the environment is formalized as a Markov Decision Process (MDP).

2.1 AEOSMP Problem

In the AEOSMP problem, a low Earth orbit (LEO) satellite observes the tasks on the Earth's surface and downloads data to ground stations. The main goal is to maximize the weighted sum of the collected and downloaded targets under complex resource constraints.

Observation targets can be specified as either points or simple polygons. A point target is a single desired location and can be captured with one image, while a polygon represents a larger area that needs to be fully captured. If the polygon's area exceeds the sensor's field of view, multiple captures are required to cover the entire area of interest (AOI). This introduces the challenge of decomposing the area into individually feasible sensor collections. Given the continuous nature of the problem, there are infinitely many ways to achieve this decomposition. Eddy et al. [7] proposed a spherical geometry-based tessellation algorithm to divide the AOI into a set of feasible tiles. This paper addresses the scheduling of tasks after decomposition, focusing on point locations and excluding stripe targets. The complete set of targets is denoted as O. The subset of the next J unimaged targets is called U, with each target in C denoted as c_j. This subset is defined by

$$C = \{c_j \in U \setminus D \mid \forall j \in [1, J)\} \tag{1}$$

where D is the set of already imaged or passed targets. At each interval, the satellite considers only the targets in C for imaging, reducing the action space to a manageable number of upcoming targets rather than the entire set O. In this work, $|C| = J = 3$ and $|O| = 135$.

The AEOS is equipped with a 3-axis attitude determination and control system (ADCS), a power management system, and a data management system. The environment and satellite are modeled using the high-fidelity Basilisk framework [8], which offers several advantages over traditional mathematical models: 1) Precise simulation of satellite orbits considering atmospheric drag and J2 perturbations, providing a realistic representation of LEO dynamics. 2) Detailed modeling of critical satellite subsystems, particularly the ADCS, which includes multiple reaction wheels and thrusters for attitude adjustments under the Modified Rodrigues Parameters (MRP) control law. 3) Facilitates the development of robust algorithms that are both theoretically sound and practically viable, enhancing the transferability from simulation to real-world implementation. 4) Integrated examination of the AEOSMP problem, considering the complex interplay between orbital mechanics, satellite subsystems, and environmental factors, thus providing a comprehensive and high-fidelity simulation environment.

2.2 MDP Formulation

An MDP is formulated to represent the AEOSMP problem. Formally, an MDP is defined by a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, R, \gamma)$, where $\mathcal{S}$ is the set of states, $\mathcal{A}$ is the set of actions, $\mathcal{P}$ is the transition probability function, R is the reward function, and γ is the discount factor.

State Space $\mathcal{S}$ contains the satellite position and velocity in the Earth-Centered, Earth-Fixed (ECEF) frame, target information, attitude rate (describes the rate of angular change in the satellite's orientation), reaction wheel speeds, remaining battery charge, eclipse indicator (a binary flag indicating whether the satellite is in Earth's shadow, affecting its solar power generation capability), remaining data storage, and downlinked data.

The action space $\mathcal{A}$ consists of charging, desaturation, downlink, and targeting images. During the charge mode, it deactivates its imaging and transmission systems and realigns its solar panels toward the sun to maximize battery recharging. In the desaturation mode, thrusters offload excess momentum from the reaction wheels, maintaining operational stability. The downlink mode is activated for data transmission when ground stations are within the specified elevation and range, allowing the satellite to send data back to Earth. Additionally, when imaging targets, the satellite adjusts its orientation to focus on specific targets c_j within the set C, gathering necessary data from these points of interest.

Given the ongoing dynamics of the AEOSMP problem, constructing an explicit transition function $\mathcal{P}$ that accurately captures state transitions using conditional probabilities proves challenging. This difficulty arises because the scheduling environment continuously evolves, making defining precise probabilities for each possible state transition hard. Basilisk [8] is employed to simulate the intricate interactions between the satellite and its environment. Utilizing the current state s_i and action a_i, the generative model $G(s_i, a_i)$ produces the subsequent state s_{i+1} and associated reward r_i. This is achieved by sampling from an underlying probability distribution and possibly integrating equations of motion or employing a hybrid approach that combines both methods. The essential factors for stochastic transitions include the distribution of the target's location, the Ground Station's location, the attitude adjustment time, the battery charge, and the data storage.

The reward function is a piecewise function based on the current state, action, and next state. The reward at step i is given by:

$$R(s_i, a_i, s_{i+1}) = \begin{cases} -1 & \text{if failure occurs} \\ \frac{1}{|I|} \sum_{j=1}^{|O|} H(d_j) & \text{if } a_i \text{ is downlink} \\ \frac{1}{|I|} H(w_j) & \text{if } a_i \text{ is image } c_j \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

Here, failure means data buffer overflow, excessive reaction wheel speeds, or battery depletion occurs. d_j and w_j are binary indicators for whether target c_j is downlinked or imaged at step i. $H(d_j)$ and $H(w_j)$ equal $\frac{1}{p_j}$ when their respective indicators are true, and 0 otherwise, where p_j is target c_j's priority. $|I|$ denotes the total number of decision intervals in the planning horizon.

3. Cluster-Enhanced Planning Strategies

We address the challenges of the AEOSMP problem by proposing two novel algorithms: the SCC-PPO Algorithm and the SCP-PPO algorithm. SCC-PPO first partitions the mission tasks into cliques — tightly connected groups of observation targets — simplifying the planning process by reducing it to manageable subproblems. It then applies PPO to optimize the sequencing of these tasks within and between cliques, enhancing overall mission efficiency. In contrast, SCP-PPO integrates clustering and planning into a single, simultaneous process. Utilizing preprocessed neighboring task information, it employs PPO to form clusters while concurrently planning the observation sequence dynamically.

3.1 SCC-PPO

For each target c_j, we calculate its distance to other targets $c_{j'}$. If the distance $l_{jj'}$ is less than a clustering radius l, they are marked as adjacent, forming an adjacency matrix:

$$\text{adj}(c_j, c_{j'}) = \begin{cases} 1 & \text{if } l_{jj'} < l \text{ and } j \neq j' \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

A graph constructed from the adjacency matrix is processed using the Bron-Kerbosch algorithm [9] to identify all maximal cliques. This algorithm maintains three sets: Q (current clique), P (potential candidates for expansion), and X (already processed vertices). It employs a recursive backtracking approach, exploring combinations and pruning branches that cannot form maximal cliques. This ensures efficient identification of all maximal target clusters in the graph. Each clique represents a cluster of closely related targets that can be observed together. The clustered tasks are then scheduled using a Proximal Policy Optimization (PPO) algorithm. The combined algorithm is shown in Algorithm 1.

The policy π_θ is optimized using the PPO objective, which comprises policy loss, value function loss, and entropy regularization:

Algorithm 1 SCC-PPO

1: Initialize clustering radii for adjacency marking and PPO parameters θ.
2: Adjacency marking: For each target c_j, compute distances to all other targets and mark adjacencies as Equation 3.
3: Clustering tasks using Bron-Kerbosch algorithm[9].
4: **for** each episode **do**
5: Observe the state s_t.
6: Select an action a_t based on the current policy π_θ.
7: Execute action a_t and observe reward r_t and new state s_{t+1}.
8: Update the policy π_θ using Eq. 4.
9: **end for**

$$L_i(\theta) = \hat{\mathbb{E}}_i \left[L_i^{CLIP}(\theta) - \alpha_1 L_i^{VF}(\theta) + \alpha_2 S[\pi_\theta](s_i) \right] \tag{4}$$

where

$$L_i^{CLIP}(\theta) = \hat{\mathbb{E}}_i \left[\min \left(r_i(\theta)\hat{A}_i, \text{clip}\left(r_i(\theta), 1-\epsilon, 1+\epsilon\right)\hat{A}_i \right) \right] \tag{5}$$

$$L_i^{VF}(\theta) = \hat{\mathbb{E}}_i \left[(V_\theta(s_i) - V)^2 \right] \tag{6}$$

$$S[\pi_\theta](s_i) = -\sum_{a \in \mathcal{A}} \pi_\theta(a|s_i) \log \pi_\theta(a|s_i) \tag{7}$$

The clipped surrogate objective, $L_i^{CLIP}(\theta)$, stabilizes policy updates by using a clipped probability ratio $r_i(\theta) = \frac{\pi_\theta(a_i|s_i)}{\pi_{\theta_{old}}(a_i|s_i)}$ to constrain update magnitudes. $\hat{A}_i$ is the advantage function, and ϵ is a hyperparameter. The value function loss, $L_i^{VF}(\theta)$, enhances state-value estimations by minimizing the squared difference between the predicted state value $V_\theta(s_i)$ and the target value V. The entropy term, $S[\pi_\theta](s_i)$, promotes exploration and prevents premature convergence, where $\pi_\theta(a|s_i)$ denotes the action probability under policy π_θ at state s_i. Hyperparameters α_1 and α_2 balance the value function loss and the entropy term.

3.2 SCP-PPO

Unlike SCC-PPO, SCP-PPO combines PPO with a task clustering algorithm to address the AEOSMP problem, as demonstrated in Algorithm 2. This hybrid approach fully exploits PPO's optimization power and clustering efficiency.

As with the SCC-PPO method, we calculate the distance between each pair of targets and mark them as adjacent if the distance is less than a threshold. Instead of clustering tasks into disjoint cliques, we mark neighbors of each target c_j.

$$\text{neighbors}(c_j) = \{c_{j'} \mid \text{adj}(c_j, c_{j'}) = 1\} \tag{8}$$

During the training process, we dynamically adjust the observation strategy based on the neighbors of each target.

4. Results and Discussion

4.1 Simulation Scenarios

Over a 3-orbit planning horizon, there are 135 imaging targets, each with assigned priorities, as shown in Fig. 2. Since the AEOS requires approximately 6 minutes to adjust its

Algorithm 2 SCP-PPO

1: Initialize threshold for adjacency marking and PPO parameters θ.
2: Adjacency marking: For each target c_j, compute distances to all other targets and mark adjacencies as Equation 3.
3: Clustering: Group adjacent targets into clusters, as Equation 8.
4: **for** each episode **do**
5: Observe the state s_t.
6: Select an action a_t based on the current policy π_θ.
7: Execute action a_t and observe reward r_t and new state s_{t+1}.
8: Dynamic Adjustment: If a target c_j is observed, mark all unobserved targets in cluster(c_j) as observed.
9: Update the policy π_θ using Eq. 4.
10: **end for**

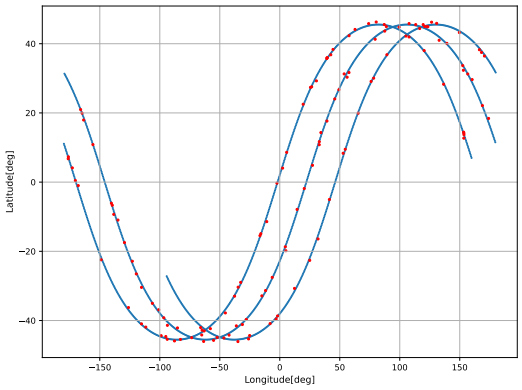

Fig. 2 Simulation scenarios. Along the flight path of the 3-orbit planning horizon, there are 135 targets available for imaging, and the planning horizon is broken into 45 discrete planning intervals, each for 6 minutes.

attitude, the planning horizon is divided into 45 decision-making intervals (i.e., $|I| = 45$), each lasting 6 minutes. The swath width of the EOS varies from 10 km to 100 km.

The algorithm's implementation incorporates carefully calibrated hyperparameters to ensure optimal performance. The learning process employs a learning rate of 0.003 and a discount factor (γ) of 0.999 to prioritize long-term reward optimization. The neural network architecture features two hidden layers, each comprising 512 units, with training conducted over ten epochs using a batch size 5000. Algorithm stability is maintained through a Generalized Advantage Estimation parameter(GAE) of 0.95 and a grad clip parameter of 0.1. The loss function coefficients are set to $\alpha_1 = 0.5$ and $\alpha_2 = 0.01$ to balance optimization objectives. The PPO clip range ϵ is set to 0.1 to prevent excessive policy updates.

4.2 Simulation Results

To evaluate the efficacy of the proposed strategies, we conducted a comprehensive comparison between SCC-PPO and SCP-PPO. The results, as illustrated in Table 1, consistently demonstrate the superiority of the SCP-PPO over SCC-PPO across different clustering radii l. Notably, the SCP-PPO method exhibited performance improvements ranging from 1.01% to 11.43% across various scenarios.

The underlying reason for this performance disparity lies in the difference between the two approaches. The SCC-PPO method often results in sub-optimal planning outcomes by segregating the clustering and planning processes. This occurs even when the initial clustering is effective, as the separation creates a disconnect between these crucial stages. Conversely, the SCP-PPO method seamlessly integrates the clustering and planning processes. This integration ensures a more holistic and balanced approach, allowing dynamic adjustments between clustering and planning decisions. Consequently, the SCP-PPO method achieves a more synergistic optimization, yielding higher overall rewards and demonstrating effectiveness.

Table 1 Rewards of algorithms over various clustering radii l.

Algs. \ l	10km	40km	70km	100km
PPO	0.4959	0.4959	0.4959	0.4959
SCC-PPO	0.6760	0.6920	0.6728	0.6732
SCP-PPO	0.6829	0.7109	0.7412	0.7502

5. Conclusion

This letter formulates the AEOSMP with a task clustering problem as an MDP and proposes two DRL-based strategies. The numerical experiments on a high-fidelity satellite simulation environment reveal that the SCP-PPO method yields higher rewards and better performance than the SCC-PPO. Specifically, the SCP-PPO method showed performance improvements ranging from 1.01% to 11.43% than the SCC-PPO. Even the SCC-PPO method performs well in initial clustering, its planning results are often suboptimal due to the separation of clustering and planning. On the other hand, the SCP-PPO method integrates these stages into a unified process. This integration allows for a more holistic and balanced approach, facilitating real-time adjustments between clustering and planning. Further work will focus on applying the proposed method to the multiple-satellite environment.

Acknowledgments

This work is supported by the National Key Research and Development Program of China (2022YFB3902801).

References

[1] X. Wang, G. Wu, L. Xing, and W. Pedrycz, "Agile earth observation satellite scheduling over 20 years: Formulations, methods, and future directions," IEEE Syst. J., vol.15, no.3, pp.3881–3892, 2020.
[2] M. Lemaître, G. Verfaillie, F. Jouhaud, J.M. Lachiver, and N. Bataille, "Selecting and scheduling observations of agile satellites," Aerospace Science and Technology, vol.6, no.5, pp.367–381, 2002.
[3] M. Stephenson and H. Schaub, "Reinforcement learning for earth-observing satellite autonomy with event-based task intervals," AAS Rocky Mountain GN&C Conference, Breckenridge, CO, 2024.
[4] A. Herrmann and H. Schaub, "A comparative analysis of reinforcement learning algorithms for earth-observing satellite scheduling," Frontiers in Space Technologies, vol.4, p.1263489, 2023.
[5] L. Dalin, W. Haijiao, Y. Zhen, G. Yanfeng, and S. Shi, "An online distributed satellite cooperative observation scheduling algorithm based on multiagent deep reinforcement learning," IEEE Geosci. Remote Sens. Lett., vol.18, no.11, pp.1901–1905, 2020.

[6] G. Wu, J. Liu, M. Ma, and D. Qiu, "A two-phase scheduling method with the consideration of task clustering for earth observing satellites," Computers & Operations Research, vol.40, no.7, pp.1884–1894, 2013.

[7] D. Eddy and M.J. Kochenderfer, "A maximum independent set method for scheduling earth-observing satellite constellations," Journal of Spacecraft and Rockets, vol.58, no.5, pp.1416–1429, 2021.

[8] P.W. Kenneally, S. Piggott, and H. Schaub, "Basilisk: A flexible, scalable and modular astrodynamics simulation framework," Journal of Aerospace Information Systems, vol.17, no.9, pp.496–507, 2020.

[9] E.A. Akkoyunlu, "The enumeration of maximal cliques of large graphs," SIAM J. Comput., vol.2, no.1, pp.1–6, 1973.