IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 31 October 2023, accepted 26 November 2023, date of publication 28 November 2023,
date of current version 6 December 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3337358

RESEARCH ARTICLE

MARL-Based Multi-Satellite Intelligent Task Planning Method

GUOHUI ZHANG, XINHONG LI, GANGXUAN HU, YANYAN LI, XUN WANG, AND ZHIBIN ZHANG

Department of Aerospace Science and Technology, Space Engineering University, Beijing 101416, China

Corresponding author: Zhibin Zhang (zhangzhibinseu@163.com)

ABSTRACT In this article, we propose a solution to multi-satellite intelligent task planning using the multi-agent reinforcement learning (MARL) method. Fristly, we have developed a multi-satellite task planning model based on the Markov game framework. Furthermore, we have computationally designed a satellite state transition function to address the task planning problem and successfully solved it using the multi-agent proximal policy optimization (MAPPO) algorithm. Our experimental results demonstrate that the MARL method exhibits remarkable convergence speed and performance, delivering significant rewards in multi-scale task planning scenarios. Consequently, it proves to be a highly suitable approach for multi-satellite intelligent task planning.

INDEX TERMS MARL, multi-satellite intelligent task planning, Markov game, MAPPO.

I. INTRODUCTION

Satellite task planning in remote sensing is a technology that rationally allocates satellite system resources and makes action sequences to optimize the execution of satellite observation tasks according to the needs of users. This process presents challenges due to the diverse and voluminous information, complex constraints, and the inherent combinatorial explosion as time progresses [1].

Satellite Earth observation technology has developed rapidly, but due to the changeable space operating environment, the change of satellite status and the increase and change of user requirements, the existing mission planning technology has posed a great challenge. It is mainly reflected in:

1) With the increasing demand for observation information services, the number of observation requirements and the complexity of observation requirements are also greatly increased. Most importantly, the timeliness of information services is also increasingly required.
2) During the actual operation of satellites, we need to consider a series of practical constraints related to the number of satellites, the number and type of loads, the amount of power of satellites, and the attitude of satellites, which are difficult to be considered and perfected in the models established in the past, and have been simplified to a certain extent.
3) During the actual operation of the satellite, it will encounter a series of emergencies, and the state of the satellite and the task will be disturbed, such as the satellite failure and the increase of emergency tasks.

Currently, this problem is typically addressed through central management and control by ground operation centers. The process involves the ground task planning system generating observation programs, transmitting them to satellites as commands, and satellites executing the tasks accordingly [2]. However, managing large-scale remote-sensing satellites in the traditional manner leads to a significant increase in command data destined for satellites, placing more pressure on ground observation and control stations. Moreover, this approach fails to handle emergency situations or adapt to complex user needs, such as unexpected resource usage, temporary equipment failures, or the inclusion of emergency tasks. Thus, the satellite task planning process must transition from periodic ground control to an autonomous pattern, such as onboard real-time response, which not only represents an inevitable trend but also fulfills the demands of users.

The advancement of artificial intelligence and aerospace technologies has enabled autonomous satellite task planning.

The associate editor coordinating the review of this manuscript and approving it for publication was Yilun Shang.

Onboard autonomous task planning serves as a critical core technology for the autonomous operation of remote-sensing satellites, acting as the ''brain'' that controls task execution. This application of onboard autonomous task planning enhances the intelligence of remote-sensing satellites, enabling them to act not only as command executors but also as task decision-makers [3]. Yan et al. [4] constructed a performance index function considering formation error cost and control input energy cost for the prescribed-time formation control problem. By developing the RL algorithm with ACI structure, an optimal prescribed-time control approach was presented, which can minimize the constructed performance index function. Chen et al. [5] propose an adaptive-estimator-based sliding-mode control protocol to solve the event-triggered connectivity-preserving consensus problem for uncertain MELSs. Using the Lyapunov method, they demonstrated the asymptotic stability of MELSs. However, current studies on applying artificial intelligence to satellite task planning are still in the early stages. Usaha and Barria [6] proposed a multi-agent planning method for earth remote-sensing spacecraft swarms, and gave the main constraints and evaluation criteria of spacecraft group planning efficiency. Through ground training and in-orbit application, satellites can efficiently complete autonomous task planning. Tinker et al. [7] developed a case-learning-based method for satellite observation task planning, using historical datasets for unsupervised learning and predicting task schedulability. Liu et al. [8] used the integrated BP neural network method to design a componentized solution architecture composed of collaborative task assignment component, task scheduling component, feature extraction component and task schedulability prediction component to predict the schedulability of observation tasks. Chen et al. [9] proposed an end-to-end framework based on deep reinforcement learning and built a neural network that introduced RNN and attention mechanism. The model regarded the neural network as a complex heuristic method and trained it using Actor-critic algorithm. Huang [10] used graph clustering to preprocess tasks for single-star task planning, regarded task decision as a continuous process, and decided how long visual time window to be divided into tasks. Finally, DDPG algorithm was used for deep reinforcement learning training. Luo et al. [11] presented a multi-satellite emergency observation mission planning method based on Transformer hierarchical prediction. The solution process of multi-satellite observation mission planning problem is decomposed into task schedulability prediction, task allocation, and optimization adjustment. Zhang et al. [12] proposed an online satellite task schedulability prediction model based on Bi-LSTM. The satellite offline task planning data were used as learning samples to train the model, and the model prediction results were of high accuracy.

Reinforcement learning has emerged as a prominent research area in artificial intelligence, focusing on learning policies to maximize rewards or achieve specific goals through agent–environment interaction [13]. Multi-agent reinforcement learning (MARL) applies the principles and algorithms of reinforcement learning to multi-agent systems. Littman [14] introduced the MARL approach in the 1990s, using Markov Decision Processes (MDP) as the framework for environment modeling. MARL provides a mathematical framework for solving various reinforcement learning problems and has become the foundation for subsequent studies [15].

In this study, we integrate MARL with satellite task planning to develop a Markov game-based model for multi-satellite intelligent collaborative task planning. This model represents the multi-satellite task planning problem as a multi-agent MDP, transforming it from a distributed optimization problem into a collaborative decision-making problem. By constructing a differential equation for the MDP and employing the multi-agent proximal policy optimization (MAPPO) algorithm, we propose a MARL-based distributed online satellite scheduling algorithm to solve the model. This method can solve the current problems and challenges well. Firstly, the framework of off-line training on ground and online execution on board can effectively meet the timeliness requirements of observation information service. Secondly, the Markov game model can be trained to solve the task planning problem, so that it can be directly solved. Finally, in the case of increasing emergency tasks, new tasks can be directly added to the task sequence to be decided for direct strategy solving.

The structure of this paper is as follows: In the first part, we analyzed the background and necessity of this research, sorted out the current research status, and finally determined the research content of this paper. In the second part, we first briefly introduce the principle of Markov game, and then define the parameters, assumptions and constraints of multi-star task planning, so as to establish a task planning model based on Markov game. Finally, we introduce the framework and principle of multi-agent near-end strategy optimization algorithm and give the basic flow of model-based programming network algorithm based on the system differential equation of Markov decision process. In the third part of the paper, we verify the proposed model and algorithm in different scales of task planning scenarios through simulation experiments. Then we compare our method with genetic algorithm and tabu search algorithm. Then, in the fourth and fifth parts of this paper, we further discuss the experimental results and the advantages of the proposed method, and finally we give the conclusions and future research directions of this paper.

II. MATERIALS AND METHOD

A. SATELLITE TASK PLANNING MODEL BASED ON MARKOV GAME

When applying reinforcement learning to the multi-satellite task planning problem, there exists a corresponding relationship between the four elements of reinforcement learning and the objects in the multi-satellite task planning problem. These elements include:

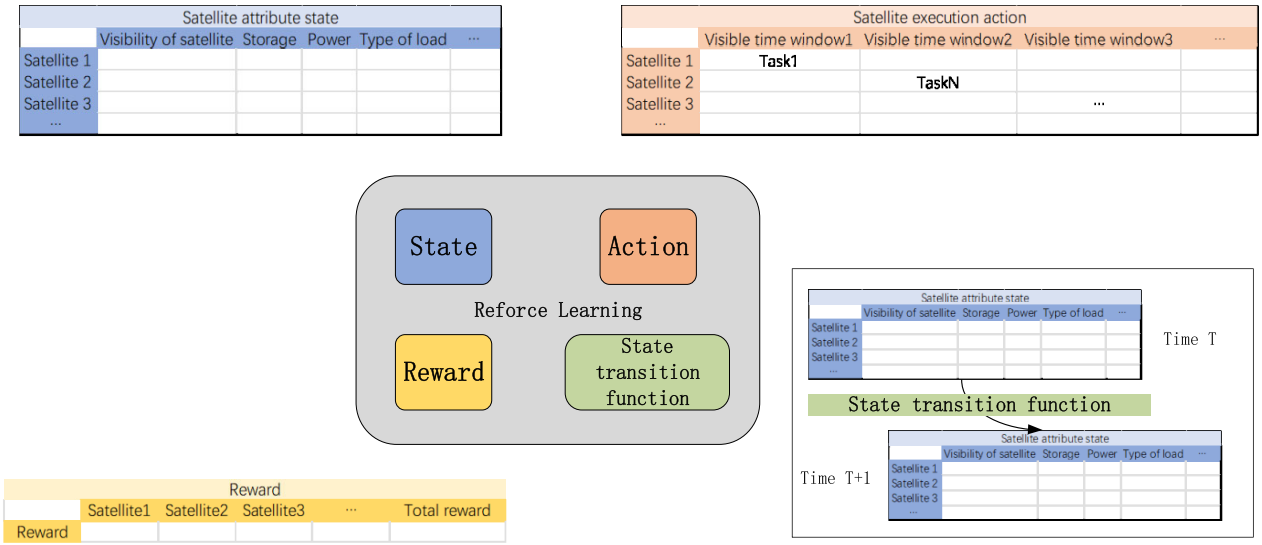

FIGURE 1. Reinforced learning framework for multi-satellite task planning.

1) "State" refers to the attribute state of a satellite, including visibility to targets, storage, electric quantity and load type, load resolution, and posture;
2) "Action" means whether a satellite executes a task within the visible time window after task planning;
3) "Reward" refers to the evaluation of the rewards obtained after the system takes the above action;
4) "State transition" refers to the change of the satellite state from that moment to the next moment after the system takes an action at the moment.

Accordingly, FIGURE 1 shows the reinforced learning framework for multi-satellite task planning.

The multi-satellite task planning process can be outlined as follows: A set of ob-servation tasks, either sent from the ground or autonomously generated by satellites, is received by the multi-satellite system in chronological order. The satellites make se-quential decisions to determine the next task to be executed. This decision-making problem can be represented as a Markov Decision Process (MDP), where each satellite, upon completing the current task, not only receives an immediate reward but also ex-periences a state transition that influences subsequent rewards. Consequently, the task planning problem is transformed into a multi-agent policy optimization problem.

B. OVERVIEW OF MARKOV GAME

The MARL environment is an MDP-based stochastic game framework [16]. A Markov game can be viewed as an extension of MDP, where multiple agents make multiple action decisions in multiple states. Each agent, based on its own state, strives to make optimal action decisions by observing the environment and predicting the actions of other agents, aiming to improve its value function.

In this study, we formally describe the Markov game $\langle N, S, A, R, P\rangle$ using a multivariate array [17]. Here, N denotes the number of agents, S represents the state space of the environment, $A = a_1 \times a_2 \times \cdots \times a_N$ denotes the joint action space of all agents, $R = r_1 \times r_2 \times \cdots \times r_N$ represents the joint reward space for all agents, and $P\left(s, a, s'\right)$ denotes the joint transition probability. The state transition process is shown in FIGURE 2.

The agent obtains the state information of the environment at time t through observation, and decides the current action according to the state information. Through the information interaction between each other, multiple agents output joint action A_t, which makes the state of the environment transition from S_t to S_{t+1}. Then the joint state transition function is as follows.

$$P\left(S, A, S'\right) : S_t \times A_t \times S_{t+1} \rightarrow [0, 1] \tag{1}$$

$[0, 1]$ in Eq. (1) represents the probability distribution of the transition of the environment state from S_t to the next state S_{t+1} given the execution of the joint action A_t.

The policy adopted by an agent is a set or distribution of probabilities, where the element $\pi\ (a \mid s) = \mathrm{P}[A_t = a | S_t = s]$ signifies the probability of taking a specific action a given the current state s. Importantly, the policy π is solely dependent on the current state and not influenced by historical information. $\pi = [\pi_1, \pi_2, \ldots, \pi_N]$ is the joint policy. Each agent policy function π_i can be expressed as:

$$\pi_i\left(S, a\right) : S_t \times a_t^i \rightarrow [0, 1] \tag{2}$$

C. DEFINITION OF PARAMETERS AND DECISION VARIABLES

In the satellite task planning process, the system focuses on planning the current observation task independently of other tasks in subsequent periods. It primarily needs to determine whether the next task should be executed and which satellite is best suited to carry it out optimally. To describe the system

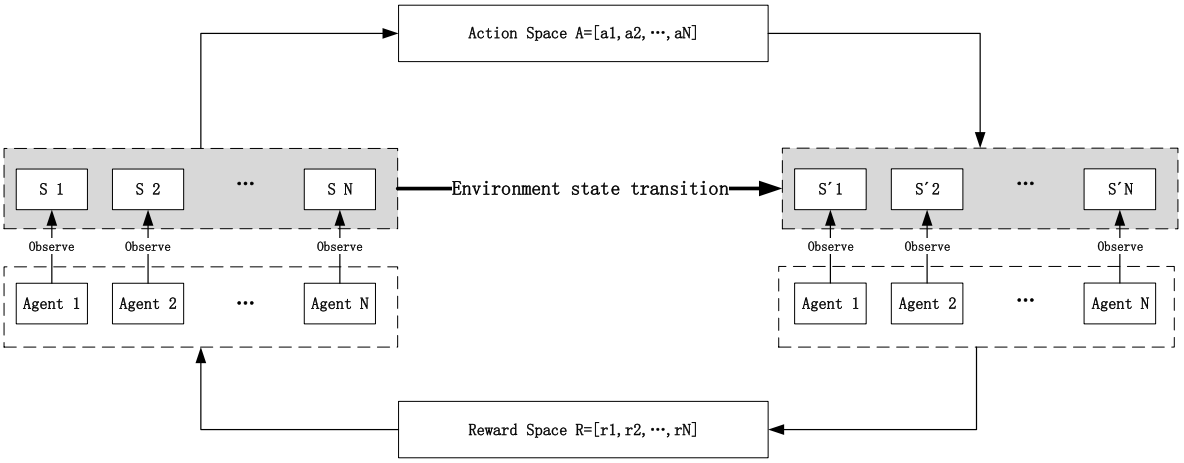

FIGURE 2. State transition process.

task planning problem, this study draws upon the sequential decision model used in decision problems [18]. The detailed process can be outlined as follows: A set of observation tasks, either sent from the ground or generated autonomously by satellites, is received by the multi-satellite system in chronological order. The satellites make sequential decisions until they identify the next task to be executed. The related factors in the problem are analyzed in the following section:

1) The observation task set is represented as $\text{Task} = \{\text{task}_1, \text{task}_2, \ldots, \text{task}_i, \ldots, \text{task}_M\}$. Upon receiving tasks from the ground command center, other satellites in the multi-satellite system, or autonomously generating tasks, the intelligent imaging satellite performs executability prediction. It generates a set of M observation tasks that can be executed by the onboard intelligent task planning system. These observation tasks arrive sequentially at specific times denoted by $A = \{A_1, A_2, \ldots, A_i, \ldots, A_M\}$, where, $A_i \in T = [T_s, T_e]$ represents the arrival time of task $task_i$. Here, T_s denotes the planned start time, and T_e denotes the planned end time. It is assumed that the set of tasks is all completed within the planned period. Each task t_i is associated with relevant information defined as $task_i = \langle P_i, Pow_i, D_i, WinS_i, WinE_i, Load_i, Res_i \rangle$. Here, P_i denotes the reward of $task_i$, Pow_i represents the power consumption during $task_i$ execution, D_i indicates the storage capacity occupied by the $task_i$, $WinS_i$ signifies the start time of the $task_i$'s time window, $WinE_i$ represents the end time of the $task_i$'s time window, $Load_i$ indicates the type of satellite payload necessary for observation $task_i$ and Res_i represents the resolution of the satellite payload necessary to observe $task_i$.
2) The satellite set is denoted as $S = \{s_1, s_2, \ldots, s_j, \ldots, s_N\}$. The intelligent task planning system comprises N satellites, and each satellite's attributes can be represented as a binary group $s_j = \langle \text{Pow}_{max_j}, Dmax_j, \text{LOAD}_j, \text{RES}_j \rangle$. Here, Pow_{max_j} denotes the maximum power that satellite s_j can utilize for executing tasks within the planned period, D_{max_j} designifies the maximum storage capacity available for satellite s_j during task execution, LOAD_j represents the type of payload carried by satellite s_j and RES_j represents the resolution of the payload carried by satellite s_j.
3) Decision-making variable: The decision-making variable $x_{i,j}\,(i \in M, j \in N)$ for task planning indicates whether task t_i is planned to be executed by satellite s_j. If the task t_i is assigned to satellite s_j, $x_{i,j} = 1$; otherwise, $x_{i,j} = 0$.

The following equation represents whether the satellite is in the state of executing task $task_i$ at any time t:

$$x_{i,t}\,(i \in M, t \in T) = \begin{cases} 1, & \text{WinS}_i < t < \text{WinE}_i \& x_{i,j} = 1 \\ 0, & else \end{cases} \tag{3}$$

D. BASIC HYPOTHESES

Compared with single-satellite task planning problem, multi-satellite task planning problem is more difficult to solve and needs to consider more constraints. It is also a NP-hard problem. Therefore, in this study, in order to highlight the key constraints in the problem, some reasonable simplification of the research content is made. Reasonable assumptions and simplifications made in this section are as follows:

1) In multi-satellite systems, satellites can communicate with each other through inter-satellite communication links or relay satellites to achieve low-delay state communication, which can ensure that satellites can observe the global state at the current time.
2) In the multi-satellite system, all satellites are imaging satellites, and there are no communication satellites or agile satellites. The payload carried by the satellite can be either an optical camera or a SAR payload.

In addition, the resolution of the payload is fixed, and the observation mode and resolution required by the observation target are fixed. Therefore, we need to consider the satellite payload and resolution constraints.

3) In the multi-satellite system, the storage space of all satellites is limited. In this paper, it is assumed that the storage space occupied by the observation activity of the satellite will not be released, so we should consider the constraint of storage space consumption.
4) In the multi-satellite system, the power value of all satellites is limited. In this paper, it is assumed that during the replanning period of the satellite, the power consumed by the satellite to perform the observation task is separate from that consumed by other satellite activities. Therefore, when the total power is fixed, we need to consider the constraint of power consumption.
5) Within the given time period $\mathrm{T} = [T_s, T_e]$, where T_s denotes the planned start time and T_e denotes the planned end time, tasks arrive sequentially in the planning sequence without any temporary emergency tasks interposed.
6) Considering non-periodic tasks, each task is considered complete when it is executed within any time window. Additionally, each task can only be executed once and not repeatedly.
7) Each satellite can only perform one task at a time.
8) Different satellites have the capability to transmit real-time data to each other. After the completion of observation tasks within the visible time window between satellites and ground stations, observation data is transmitted to the ground stations, irrespective of data transmission planning.

E. CONSTRAINT ANALYSIS

Based on the defined parameters, variables, and basic hypotheses, the following constraints on satellites executing observation tasks can be analyzed. These constraints include task constraints, satellite constraints, ground station constraints, and environmental constraints. They are described as follows:

1) Temporal constraint: Each satellite can only execute one task at a time, and there should be no temporal overlap between two consecutive tasks executed by any satellite. This constraint can be represented by the following equation:

$$\mathrm{WinS}_{i+1} - \mathrm{WinE}_i \geq 0 \tag{4}$$

2) Task uniqueness constraint: Each task can be executed by at most one satellite. The sum of decision variables for a task across all satellites should be less than or equal to 1. Mathematically, it can be expressed as follows:

$$\sum_{i=1}^{M} x_{i,j} \leq 1, \quad j \in N \tag{5}$$

3) Energy constraint: The total energy consumed by a satellite to execute all tasks must not exceed its maximum energy capacity. This constraint can be formulated as follows:

$$\sum_{i=1}^{M} x_{i,j} \times \mathrm{Pow}_i \leq \mathrm{Pow}_{max_j}, \quad j \in N \tag{6}$$

4) Storage constraint: The storage space occupied by the results of observation tasks should not exceed the maximum storage capacity of each satellite. This constraint can be defined as follows:

$$\sum_{i=1}^{M} x_{i,j} \times D_i \leq D_{max_j}, \quad j \in N \tag{7}$$

5) Payload constraint: The type of payload required for the observation task must be the payload carried by the satellite in order to perform the observation task. This constraint can be expressed as:

$$\mathrm{Load}_i = \mathrm{LOAD}_j, \quad i \in M,\ j \in N \tag{8}$$

6) Payload resolution constraint: The resolution of the payload required for the observation task must be greater than that of the payload carried by the satellite. Only in this way can the observation task be completed if the imaging results of the observation target meet the requirements. The expression for this constraint is as follows:

$$\mathrm{Res}_i \geq \mathrm{RES}_j, \quad i \in M,\ j \in N \tag{9}$$

F. TASK PLANNING MODEL BASED ON MARKOV GAME

For the intelligent task planning system comprising N satellites, the Markov game model can be expressed as the following tuple [19]:

$$MSMDP = \langle s_1, s_2, \ldots, s_N, a_1, a_2, \ldots, a_N, r_1, r_2, \ldots, r_m, P \rangle \tag{10}$$

where, s denotes the working state of an intelligent satellite. The work state of the j-th satellite at the time $\mathrm{t} = A_i$ is defined as $s_j = \langle t_{free}, \mathrm{Pow}_t, D_t, \mathrm{LOAD}_t, \mathrm{RES}_t \rangle$.

t_{free} denotes whether at the time $\mathrm{t} = A_i$, the satellite s_j is in idle state; $Pow_{t,j}$ denotes the current available electric quantity of the j-th satellite at the time $\mathrm{t} = A_i$; $D_{t,j}$ denotes the current available storage capacity of the j-th satellite at the time $\mathrm{t} = A_i$.

a denotes the action of each intelligent satellite. In the multi-satellite task planning process, the action denotes whether the satellite executes the current incoming task.

$$a^i = x_{i,j} = \pi_j(\mathrm{task}_i) = \begin{cases} 1, & task_i \ accepted \\ 0, & task_i \ rejected \end{cases} \tag{11}$$

$\pi_j(\mathrm{task}_i)$ denotes the planning policy when the j-th satellite executes the task t_i. The action decision of the entire system is the set of the decisions made by the N-th satellite.

$$a = [a_1, a_2, \ldots, a_j, \ldots, a_N] \tag{12}$$

r denotes the reward of task execution under the policy $\pi_j(t_i)$. For the single satellite s_j, the reward function is $r_i = x_{i,j} \times P_i$. Accordingly, the total expected reward function of a single satellite is defined as follows:

$$V_\pi(A_i) = x_{i,j} \times P_i + E\left(\sum_{A_k \in A}^{T_E} x_{k,j} P_k\right), \quad A_{k>A_i} \tag{13}$$

In particular, when $A_i = 0$, the decision-making policy π_j is used across the entire decision-making period, to obtain the expected reward:

$$V_\pi(0) = E\left(\sum_{A_i=0}^{T_E} x_{i,j} P_i\right) \tag{14}$$

The total reward function of the entire system is defined as follows:

$$V_i\left(s, \pi_1, \ldots, \pi_j, \ldots, \pi_N\right) = \sum_{j=1}^{N} V_\pi(0) \tag{15}$$

In the Markov model, if the expected reward $V_j\left(\pi_1^*, \ldots, \pi_j^*, \ldots, \pi_N^*\right)$ of each agent meets the following condition under the combined policy $\left(\pi_1^*, \ldots, \pi_j^*, \ldots, \pi_N^*\right)$:

$$V_j\left(\pi_1^*, \ldots, \pi_j^*, \ldots, \pi_N^*\right) \geq V_j\left(\pi_1^*, \ldots, \pi_j, \ldots, \pi_N^*\right), \quad \forall \pi_j \in \Pi_j, j = 1, \ldots, N \tag{16}$$

So we can say that under this strategy, the system reaches Nash equilibrium [20].

G. STATE TRANSITION FUNCTION FOR SATELLITE TASK PLANNING

The system state at time $t = A_i$ is defined as $s = \langle s_1, s_2, \ldots, s_N \rangle$. When the decision $a = \langle a_1, a_2, \ldots, a_N \rangle = \pi(\text{task}_i)$ is implemented at the time $t = A_i$, the state-action function of the MDP can be defined as follows:

$$Q^\pi(s, a) = R(s, a) + \gamma \sum_{s' \in s} P\left(s, a, s'\right) V^\pi(s) \tag{17}$$

where, $P\left(s, a, s'\right)$ denotes the state transition function of the MDP.

For any satellite s_j, the following functional relations exist between the state s at time $t = A_i$, the action a_j^i taken according to the current policy, and state $t = A_{i+1}$ at the next decision-making time s':

$$t'_{free} = \begin{cases} a_j^i, & \text{WinS}_{i+1} - \text{WinE}_i < 0 \\ 0, & \text{WinS}_{i+1} - \text{WinE}_i \geq 0 \end{cases} \tag{18}$$

$$\text{Pow}'(t = A_{i+1}) = \text{Pow}'(t = A_i) - a_j^i \times \text{Pow}_i \tag{19}$$

$$D'(t = A_{i+1}) = D'(t = A_i) - a_j^i \times D_i \tag{20}$$

Therefore, the state transition function of the system at the time $t = A_i$ is as follows:

$$\begin{cases} t'_{free} = \begin{cases} a_j^i, & \text{WinS}_{i+1} - \text{WinE}_i < 0 \\ 0, & \text{WinS}_{i+1} - \text{WinE}_i \geq 0, \end{cases} \\ \text{Pow}'(t = A_{i+1}) = \text{Pow}'(t = A_i) - a_j^i \times \text{Pow}_i \\ D'(t = A_{i+1}) = D'(t = A_i) - a^i \times D_i \end{cases} \tag{21}$$

TABLE 1. Summary of symbols and meanings.

Symbol	Meaning
i	Index of the ith observation task
j	Index of the jth satellite
M	Total number of observation task
N	Total number of satellites
A_i	Arrival time of the ith task
P_i	Profit of the ith task
Pow_i	Power consumption of the ith task
D_i	Storage consumption of the ith task
$\text{WinS}_i, \text{WinE}_i$	Start and end time of the ith task if accept
Load_i	Observation payload requirement of the ith task
Res_i	Observation payload resolution requirement of the ith task
Pow_{max_i}	Total power carried by the jth satellite
D_{max_i}	Total storage space of the jth satellite
LOAD_j	Type of payload carried by the jth satellite
RES_j	Resolution of the payload carried by the jth satellite
t_{free}	Current operating status of the satellite
a^i	Current action of the jth satellite
$x_{i,j}$	Decision variable
π_j	Task execution policy for the jth satellite
V_π	Total reward of task based on policy π

The satellite state transition function is a model that can predict the next moment state of the real environment, using the next moment state obtained by the transfer function as a label and using stochastic gradient descent for supervised learning. The input is the state information of the satellite and the mission at the current moment and the joint action space of the satellite. The output state information at the next moment can replace the real environment information and be put into the experiential return pool. Then, the reinforcement learning model of task planning is trained by random sampling.

Due to the difficulty in obtaining satellite operation data and the small amount of simulation data, the state transition function data enhancement strategy proposed in this paper can solve these problems.

This section involves many parameters and symbols. In order to facilitate readers' reading, symbols and meanings are summarized as shown in Table 1.

H. MARL-BASED ONLINE SATELLITE PLANNING ALGORITHM

MAPPO [21] is a mature multi-agent reinforcement learning algorithm, which solves the task planning problem through an optimal learning policy. Compared with traditional reinforcement learning algorithms, MAPPO can train multiple agents simultaneously and address non-cooperative tasks (i.e., possible competition between different agents) in the task planning problem. It is a parallel and distributed algorithm, which can significantly improve learning efficiency through multi-agent parallel training. Moreover, MAPPO incorporates the proximal policy optimization technique, which reduces bias

in policy optimization to optimize the learning policy effectively.

MAPPO algorithm is a strategy gradient algorithm based on AC framework, which uses random gradient ascent to optimize the objective function. When the probability distribution of action a_t is output under the state s_t, the policy network outputs π_θ as a probability distribution, and then samples continuous actions based on this distribution. Therefore, the output actions may be different even under the same state, thus possessing the ability to explore the environment.

For the multi-satellite mission planning environment, MAPPO algorithm enables the agent to learn a strategy to maximize the return expectation. The algorithm improves the acquisition of observation benefits by optimizing the strategic network Actor. For any agent satellite s_j, the actor network can be represented as π_θ^j and $\pi_\theta^j(old)$, the Critic network can be represented as V_ω^j, and the objective function can be represented as:

$$\hat{J}(\theta) = \hat{E}_t\left[\frac{\pi_\theta^j}{\pi_\theta^j(old)}\hat{A}(s_t, a_t)\right] = \hat{E}_t\left[\sigma(\theta)\hat{A}(s_t, a_t)\right] \tag{22}$$

where $\sigma(\theta)$ is the ratio of the current policy to the old policy, and $\hat{A}(s_t, a_t)$ is the advantage function estimated using generalized advantage. In order to improve the learning efficiency and training stability, MAPPO adopts clip method to limit the strategy update within the range of $[1-\varepsilon, 1+\varepsilon]$. Under this method, the loss function of the agent is as follows:

$$\hat{J}(\theta) = \hat{E}_t\left[min\left(\sigma(\theta)\hat{A}(s_t, a_t), clip(\sigma(\theta), 1-\varepsilon, 1+\varepsilon)\hat{A}(s_t, a_t)\right)\right] \tag{23}$$

Formula 23 is the update mode of the Actor network. For the Critic network, to accurately evaluate the current policy, the network updates the network parameters by minimizing the loss function. The loss function is as follows:

$$\hat{J}(\omega) = \hat{E}_t\left[\left(\hat{A}(s_t, a_t) + V_\omega^j(old) - V_\omega^j\right)^2\right] \tag{24}$$

The algorithm is divided into two parts: distributed planning and centralized execution. The centralized execution stage can be repeated on the star to reduce the training time. In addition, the algorithm can provide real-time feedback to the satellite observation action, and after the total return is obtained by accumulating the current return, the decision-making strategy can be adjusted by evaluating the cumulative return to optimize the overall return. In terms of timeliness and profitability, the proposed algorithm is suitable for multi-satellite mission planning.

FIGURE 3 shows the architecture of the MAPPO algorithm:

The MAPPO algorithm employs an Actor-Critic architecture with N agents [22], representing the N satellites considered in this study. Each agent consists of an Actor network and a Critic network. The agents operate in a central learning and distributed execution fashion. During training, the Actor network takes the agent's current state as input and produces the corresponding action. The structure of the execution network is shown in FIGURE 4. The Critic network, receiving the global state information and the actions of all agents, evaluates the current policy and generates a value function. This framework enables collaborative learning and optimization of the system's performance.

TABLE 2. Actor network parameters.

Name	In_dim	Out_dim	Activation Function
FC1	48	256	relu
FC2	256	256	relu
Output	256	2	softmax

TABLE 3. Critic network parameters.

Name	In_dim	Out_dim	Activation Function
FC1	75	256	relu
FC2	256	256	relu
Output	256	1	/

The Actor network consists of two full connection layers and one output layer. Table 2 describes the network parameters of each layer.

During the training process, the Critic network takes as input the system's state vector and the actions generated by the Actor network. Its output is the evaluation function that assesses the effectiveness of the current policy. Similar to the Actor network, the Critic network consists of two full connection layers and one output layer. Table 3 describes the network parameters of each layer.

The basic procedure of the MAPPO algorithm is described by the pseudocodes in Table 4:

III. RESULTS

A. SETTING AND TRAINING OF SIMULATION ENVIRONMENT PARAMETERS

In the simulation experiment, the algorithm framework used was TensorFlow 2.0 based on Python 3.11, with Anaconda 3.5.2 as the development environment. The hardware setup consisted of an Intel Xeon®W-2102 CPU and 16 GB of memory. Due to the unavailability of real data sets for satellites and tasks, a scenario simulation program was developed using a combination of STK software and MATLAB language. The simulation scenario involved a Walker constellation observation satellite located at a 600 km altitude in a sun-synchronous orbit. The observation targets were randomly generated within a latitude range of $-60°$ to $60°$. Other scenario parameters are listed in Table 5.

Table 6 lists the hyper-parameters in the training process:

B. ANALYSIS OF TRAINING RESULTS

During the scenario simulation, 800 observation tasks, along with their time windows and rewards, were randomly

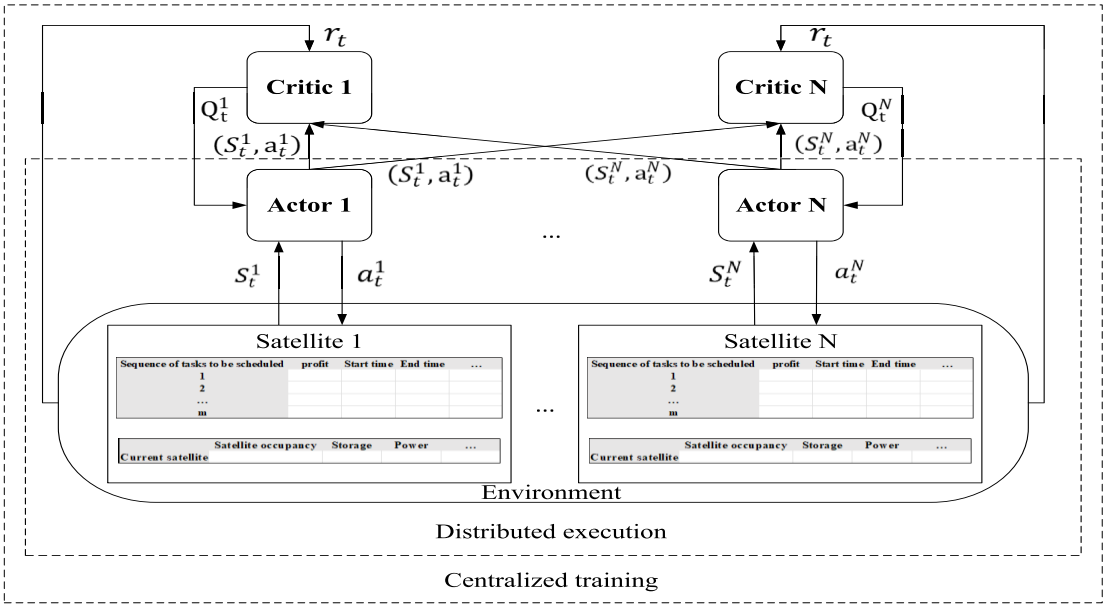

FIGURE 3. Basic architecture of the MAPPO algorithm.

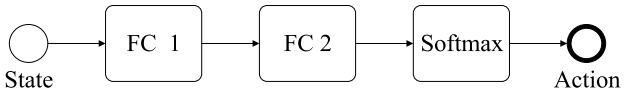

FIGURE 4. Actor network structure.

TABLE 4. Pseudocodes of the MAPPO algorithm.

Table 4 Pseudocodes of the MAPPO algorithm
Initialized parameters $\boldsymbol{Q}$ and $\boldsymbol{\pi}$, neural network parameters $\boldsymbol{\omega}$ and $\boldsymbol{\theta}$
Initialized memory pool $\boldsymbol{D} = \{\}$
while $\boldsymbol{step} < \boldsymbol{step}_{max}$ do
For $\boldsymbol{t} = \boldsymbol{1, 2, \ldots, T}$
Each agent $\boldsymbol{j}$ executes the current policy function $\boldsymbol{\pi}$, to generate the action $\boldsymbol{a_t^j}$
Execute the action $\boldsymbol{a_t^j}$, calculate the reward, $\boldsymbol{r_t^j}$ and update the state $\boldsymbol{s_{t+1}^j}$
Store $\boldsymbol{s_t^j}$, $\boldsymbol{a_t^j}$, $\boldsymbol{r_t^j}$, $\boldsymbol{s_{t+1}^j}$ into the memory pool D
End for
For $\boldsymbol{j} = \boldsymbol{1, 2, \ldots, N}$
Extract sample data from D
Calculate the gradient $\Delta\boldsymbol{\theta^j}$ and $\Delta\boldsymbol{\theta\omega^j}$ of the Actor network parameters $\boldsymbol{\theta}$ **and** $\boldsymbol{\omega}$
Update the parameter $\boldsymbol{\theta}$ using the gradient descent method, and update the parameter $\boldsymbol{\omega}$ using the gradient ascent method
End for
Clear the memory pool D
End while

generated. FIGURE 5 shows the training result curve of the MAPPO algorithm.

TABLE 5. Scenario parameters for task planning.

Parameters	Value
Simulation time	2021/06/08 16:00 - 2021/06/09 16:00
Number of satellites	10
Number of targets	800
Profit	1-10
Total storage (Gb)	360
Total power (J)	140,000

TABLE 6. Hyper-parameters in the training process.

Parameters	Value
Training threads	10
Learning rate	0.0005
ppo_epoch	3
Entropy coefficient	0.01
Reward loss coefficient	0.5
Discount coefficient $\boldsymbol{\gamma}$	0.99
Number of training steps	1,000

The convergence of the algorithm was verified by analyzing the reward punishment of uncompleted tasks during the training process. As the number of training rounds increased, the reward punishment gradually decreased and eventually converged. This indicates that the total reward obtained by the satellite increased and stabilized during the training process, and the learned decision-making policy became more stable.

To evaluate the algorithm's performance, the number of observation tasks accepted within one planning period and

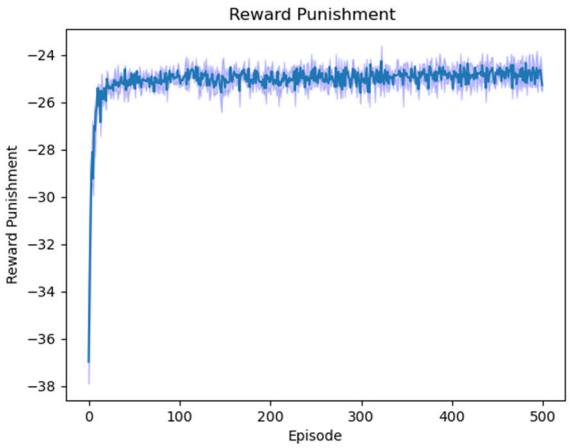

FIGURE 5. Reward punishment of training.

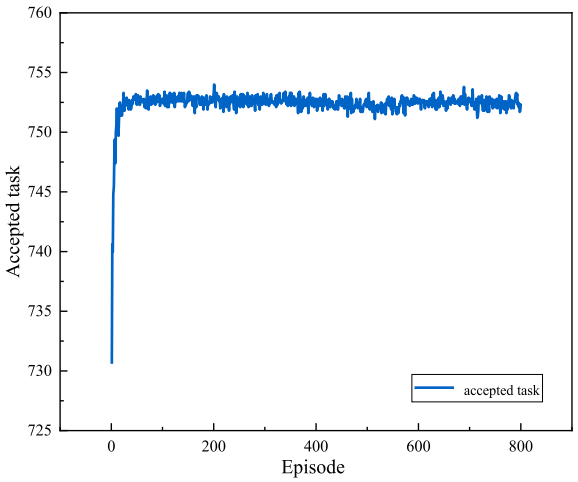

FIGURE 6. Number of observation tasks accepted by decisions.

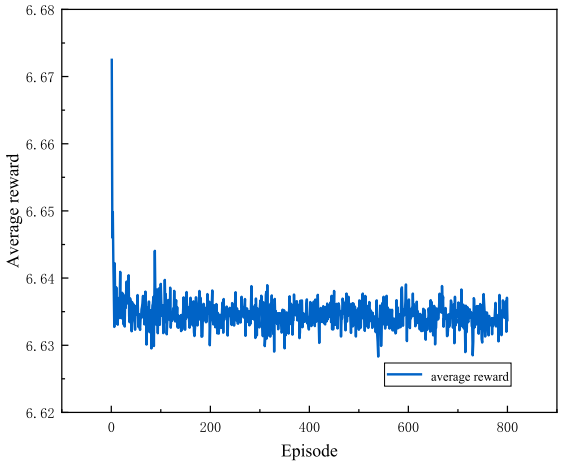

FIGURE 7. Average reward of observation tasks.

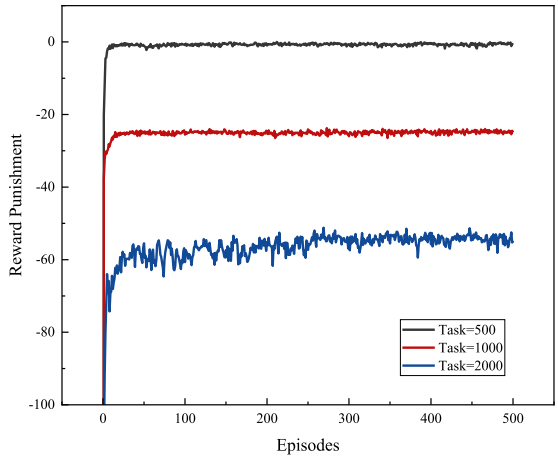

FIGURE 8. Reward punishment under different task scales.

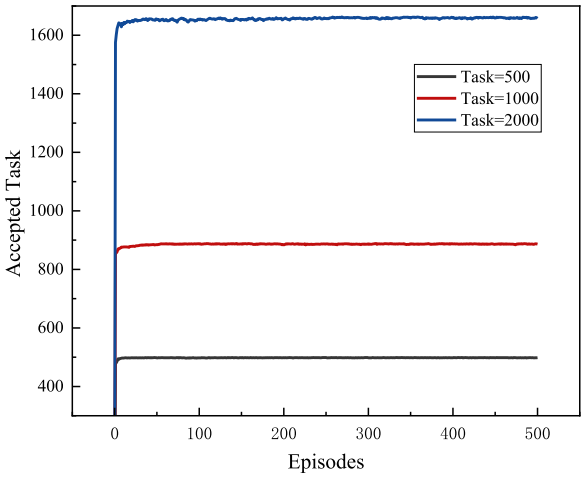

FIGURE 9. Number of accepted observation tasks on different scales.

the average reward of the accepted observation tasks were calculated.

As the training progressed, the number of accepted observation tasks increased and stabilized, as shown in FIGURE 6 and FIGURE 7. After 800 scenarios, the number of accepted tasks reached 753, an increase of 23 compared to the initial stage of training. The task execution rate improved by 3.15%. The average reward of the accepted observation tasks also tended to stabilize with an increasing number of training rounds.

C. TRAINING RESULTS AT DIFFERENT SCALES

Furthermore, the MARL algorithm's performance in task planning was evaluated on different scales by considering multi-satellite task planning cases with 500, 1,000, and 2,000 observation tasks. The simulation results demonstrated high convergence performance across different task scales, as shown in FIGURE 8, FIGURE 9 and FIGURE 10. The MARL algorithm was effective in multi-satellite intelligent task planning, as it achieved high total rewards, accepted a significant number of observation tasks, and maintained a stable average reward.

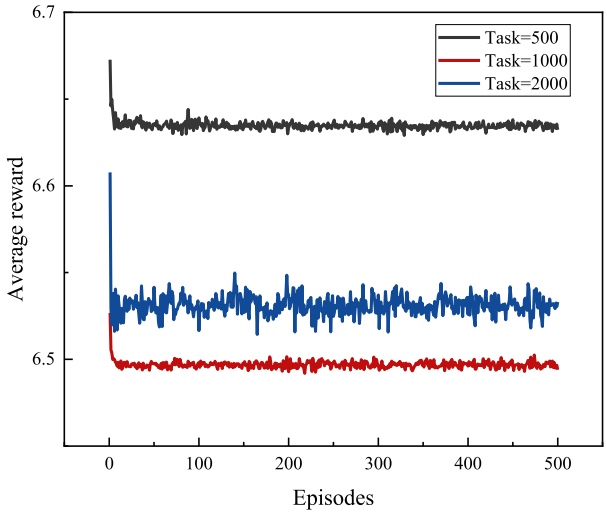

FIGURE 10. Average reward of observation tasks on different scales.

D. COMPARISON OF RESULTS

In order to further evaluate the performance of the proposed method, we compare the performance of the proposed algorithm with that of traditional multi-satellite task planning optimization algorithms. Genetic algorithm and tabu search algorithm are the most widely used optimization algorithms. The experiments in this section are verified in two different scenarios: 10 satellites with 800 observation missions and 10 satellites with 1000 observation missions. Satellite and mission attribute Settings are the same. The main parameters of genetic algorithm and tabu search algorithm are shown in Table 7 and Table 8:

TABLE 7. Genetic algorithm parameters.

Parameters	Value
Population size	50
Population crossover rate	0.8
Population variation rate	0.2
Maximum number of generations	100

TABLE 8. Tabu search algorithm parameters.

Parameters	Value
Tabu search length	The number of unscheduled tasks
Domain solution scale	5
Tabu search maximum algebra	120
Early termination mark	Task completion rate≥98%

We evaluated the performance of the compared algorithms mainly from the total reward, the number of tasks completed, the average revenue and the solving time. The experimental comparison results are shown in Table 9 and Table 10:

TABLE 9. Performance comparison (10 satellites with 800 observation tasks).

Name	the total reward	the number of tasks completed	the average revenue	the solving time
MAPPO	5003	753	6.635	43.6
GA	4647	702	6.413	121.38
TS	4843	726	6.425	90.03

TABLE 10. Performance comparison (10 satellites with 1000 observation tasks).

Name	the total reward	the number of tasks completed	the average revenue	the solving time
MAPPO	5860	903	6.497	48.91
GA	5303	862	6.152	152.38
TS	5360	866	6.174	109.03

From the above comparison, it can be seen that compared with traditional optimization algorithms, the reinforcement learning algorithm proposed in this paper can enable the agent to select the task with less resource consumption and high observation return through multiple trial and error learning training, so that a higher total task return can be obtained. In addition, the algorithm proposed in this paper can be trained several times in the early stage, and after network training, the time used to complete a task planning will be greatly shortened. Due to the large scale of solving, the solving time of genetic algorithm and tabu search algorithm will also be greatly increased, and it is easy to fall into local optimal in the optimization process.

FIGURE 11 illustrates the task planning results of 10 satellites with 800 observation tasks. The tasks corresponding to the satellites numbered 0 represent the observation tasks that were not performed due to constraint conflicts.

IV. DISCUSSION

Aiming at the problem of multi-satellite mission planning, a multi-satellite task planning model based on Markov game is established. Under the constraints of satellite resources, task time window and satellite load, the total observation reward of the mission is taken as the optimization objective, and the MAPPO algorithm is used to solve the strategy.

Firstly, we conducted experiments to verify the proposed method in different scale task planning scenarios. From the experimental results, the algorithm can converge effectively and quickly when dealing with the multi-satellite mission planning problem of 10 satellites and 500, 800, 1000 and 2000 missions respectively. From the perspective of task observation reward and average observation reward, the training of the algorithm will make the agent update the strategy and select the observation task with less resource occupation and higher reward, so that the maximum observation reward

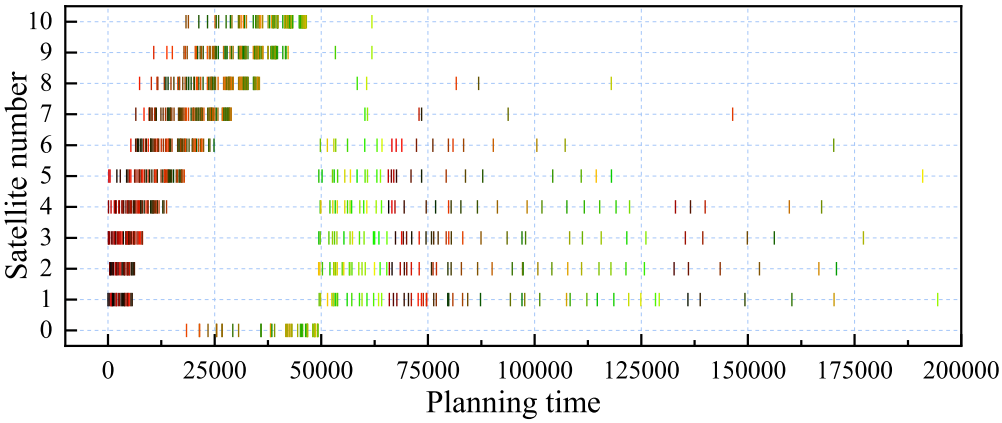

FIGURE 11. A planning results of 800 tasks by 10 satellites.

can be obtained while reducing resource consumption as much as possible.

Then, in the comparison experiment with traditional algorithms, we selected genetic algorithm and tabu search algorithm, and verified and analyzed the three algorithms in two different scale scenarios. From the four evaluation indexes of total task observation income, number of accepted tasks, average task observation income, and solving time, the algorithm proposed in this paper is superior to the traditional optimization algorithm.

From the perspective of the existing research and the realistic mission planning process of satellites, the solution of multi-satellite mission planning problems is mostly to establish a simple constraint satisfaction model, solve the problem by selecting a classical optimization algorithm, and obtain the action command. The satellite ground control center sends the action instructions corresponding to the observation task to the satellite through the telemetry and remote control system, so that the controlled object can perform a certain action to complete the observation task. However, this method is slow, inefficient and subject to various restrictions. The method proposed in this paper can improve the autonomy of satellites in the planning process and improve the efficiency of task execution, aiming at the development trend of intelligent planning. The differences and specific advantages between the method and the traditional method are as follows:

1) The traditional algorithm is embedded in the ground decision system, which has long solving time, slow speed and large calculation scale, so it is difficult to be built into the satellite to realize the satellite online mission planning. The method proposed in this paper is suitable for distributed satellite systems, where each satellite is an agent with autonomy in decision-making tasks.
2) The traditional algorithms only carry out a random search after setting the constraint rules, and the setting of the rules has a great influence on the results. The method based on reinforcement learning proposed in this paper is a process of trial and error learning, which can be trained to obtain better strategies.
3) The method proposed in this paper needs to be trained with a large amount of data in advance, which can be completed off-line on the ground. After the training, the single online decision solving time on board will be greatly shortened.

V. CONCLUSION

By combining the multi-satellite intelligent task planning problem with the MARL algorithm, this study presented a Markov-game-based multi-satellite intelligent task planning model, transforming the satellite decision problem into a Markov decision problem. The model-driven MAPPO algorithm was utilized to effectively solve the problem. Experimental results demonstrate that the MARL algorithm provides high convergence speed, rewards, and training efficiency for the task planning problem. It can be effectively applied to task planning scenarios with different scales, delivering excellent decision-making performance. Thus, the MARL-based multi-satellite intelligent task planning method can effectively support onboard intelligent task decision-making in multi-satellite systems.

In this study, some simplifications were made to account for satellite constraints. Future research can focus on analyzing real satellite constraints and refining the satellite state transition function in the task planning process, enabling direct application to onboard task planning. Additionally, investigating how the MARL-based multi-satellite intelligent task planning method can dynamically adapt to changing conditions, such as variations in satellite resources, environmental constraints, and observation tasks, would further enhance the capabilities and applicability of the proposed method in real-world scenarios.

REFERENCES

[1] W. Jun et al., "An approach for multiobjective uniting imaging scheduling of earth observing satellites," *J. Astronaut.*, 2007, doi: 10.1016/S1874-8651(08)60023-X.
[2] R. He and Y. Tan, "Solving parallel machine scheduling problems with time windows using constraint programming and tabu search," in *Proc. 4th Int. Conf. Syst. Sci. Syst. Eng. (ICSSSE)*, 2003.
[3] S. G. Ungar, J. S. Pearlman, J. A. Mendenhall, and D. Reuter, "Overview of the earth observing one (EO-1) mission," *IEEE Trans. Geosci. Remote Sens.*, vol. 41, no. 6, pp. 1149–1159, Jun. 2003.

[4] Y. Zhang, M. Chadli, and Z. Xiang, "Prescribed-time formation control for a class of multi-agent systems via fuzzy reinforcement learning," *IEEE Trans. Fuzzy Syst.*, early access, May 18, 2023, doi: 10.1109/TFUZZ.2023.3277480.
[5] C. Chen, W. Zou, and Z. Xiang, "Event-triggered consensus of multiple uncertain Euler–Lagrange systems with limited communication range," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 53, no. 9, pp. 5945–5954, Sep. 2023.
[6] W. Usaha and J. A. Barria, "Reinforcement learning for resource allocation in LEO satellite networks," *IEEE Trans. Syst. Man, Cybern. B, Cybern.*, vol. 37, no. 3, pp. 515–527, Jun. 2007.
[7] P. Tinker, J. Fox, C. Green, D. Rome, K. Casey, and C. Furmanski, "Analogical and case-based reasoning for predicting satellite task schedulability," in *Proc. Case-Based Reasoning Res. Develop., 6th Int. Conf. Case-Based Reasoning*, Chicago, IL, USA, Aug. 2005, pp. 566–578.
[8] S. Liu, G. Q. Bai, and Y. W. Chen, "Schedulability prediction method for imaging tasks of earth observation network," *J. Astronaut.*, vol. 36, no. 5, pp. 583–588, 2015.
[9] M. Chen, Y. Chen, Y. Chen, and W. Qi, "Deep reinforcement learning for agile satellite scheduling problem," in *Proc. IEEE Symp. Ser. Comput. Intell. (SSCI)*, Xiamen, China, Dec. 2019, pp. 126–132.
[10] Y. Huang, Z. Mu, S. Wu, B. Cui, and Y. Duan, "Revising the observation satellite scheduling problem based on deep reinforcement learning," *Remote Sens.*, vol. 13, no. 12, p. 2377, Jun. 2021.
[11] Z. Luo, C. Du, H. Chen, S. Peng, and J. Li, "Multi-satellite scheduling approach for emergency scenarios based on hierarchical forecasting with transformer network," *Acta Aeronauticaet Astronautica Sinica*, vol. 42, no. 4, p. 524721, 2021.
[12] G. Zhang, X. Li, X. Wang, Z. Zhang, G. Hu, Y. Li, and R. Zhang, "Research on the prediction problem of satellite mission schedulability based on bi-LSTM model," *Aerospace*, vol. 9, no. 11, p. 676, Nov. 2022.
[13] Y. Sun, L. Cao, X. Chen, Z. Xu, and J. Lai, "Overview of multi-agent deep reinforcement learning," *Comput. Eng. Appl.*, vol. 56, no. 5, pp. 13–24, 2020, doi: 10.3778/j.issn.1002-8331.1912-0100.
[14] M. L. Littman, "Markov games as a framework for multi-agent reinforcement learning," in *Proc. 11th Int. Conf. Mach. Learn.* New Brunswick, NJ, USA: Rutgers Univ., Jul. 1994, pp. 157–163, doi: 10.1016/B978-1-55860-335-6.50027-1.
[15] W. Du and S. F. Ding, "Review of multi-agent reinforcement learning," *Comput. Sci.*, vol. 46, no. 8, pp. 1–8, 2019.
[16] G. Wen, T. Yang, J. Zhou, J. Fu, and L. Xu, "Reinforcement learning and adaptive/approximate dynamic programming: A survey from theory to applications in multi-agent systems," *Control Decis.*, vol. 38, no. 5, pp. 1200–1230, 2023.
[17] J. R. Luo et al., "Study progress in multi-agent game learning," *Syst. Eng. Elect.*, pp. 1–34.
[18] S. Peng, H. Chen, C. Du, J. Li, and N. Jing, "Onboard observation task planning for an autonomous Earth observation satellite using long short-term memory," *IEEE Access*, vol. 6, pp. 65118–65129, 2018.
[19] H. Wang, Z. Yang, W. Zhou, and D. Li, "Online scheduling of image satellites based on neural networks and deep reinforcement learning," *Chin. J. Aeronaut.*, vol. 32, no. 4, pp. 1011–1019, Apr. 2019.
[20] S. Hoda, A. Gilpin, and P. E. Javier, "A gradient-based approach for computing Nash equilibria of large sequential games," 2007.
[21] C. Yu, A. Velu, E. Vinitsky, J. Gao, Y. Wang, A. Bayen, and Y. Wu, "The surprising effectiveness of PPO in cooperative, multi-agent games," 2021, *arXiv:2103.01955*, doi: 10.48550/arXiv.2103.01955.
[22] L. Engstrom, A. Ilyas, S. Santurkar, D. Tsipras, F. Janoos, L. Rudolph, and A. Madry, "Implementation matters in deep policy gradients: A case study on PPO and TRPO," 2020, *arXiv:2005.12729*, doi: 10.48550/arXiv.2005.12729.

GUOHUI ZHANG was born in Hebei, China, in 1995. He received the B.S. degree in energy and power engineering from Xi'an Jiaotong university, Xi'an, China, in 2018. He is currently pursuing the Ph.D. degree in aerospace science and technology with Space Engineering University. His research interests include space mission analysis and design and modular reconfigurable spacecraft.

XINHONG LI was born in 1972. He received the Ph.D. degree from Space Engineering University, Beijing, China. He is currently a Professor with Space Engineering University. He has published more than 30 articles and more than three monograph publications. His research interests include spacecraft design and application and space mission analysis. He received the National Science and Technology Progress Award.

GANGXUAN HU received the B.S. degree from the School of Mechanical Engineering, Xi'an Jiaotong University, Xi'an, China, in 2019. He is currently pursuing the Ph.D. degree with the Department of Aerospace Science and Technology, Space Engineering University, Beijing. His research interests include space mission analysis and design and space robotics.

YANYAN LI was born in 1985. She received the Ph.D. degree from the Beijing Institute of Technology, Beijing, China. She is currently an Associate Professor with Space Engineering University, China. Her research interests include orbital transfer optimization and spacecraft attitude dynamics and control.

XUN WANG was born in 1989. He received the Ph.D. degree from the National University of Defense Technology, Changsha, China. He is currently an Associate Professor with Space Engineering University, China. His research interests include spacecraft on-orbit servicing, spacecraft relative dynamics and control, and spacecraft intelligent control.

ZHIBIN ZHANG was born in 1995. He received the Ph.D. degree from Space Engineering University, Beijing, China. He is currently a Lecturer with Space Engineering University. His research interests include space mission analysis, spacecraft intelligent control, and reinforcement learning.