symmetry

MDPI

Article

Satellite Autonomous Mission Planning Based on Improved Monte Carlo Tree Search

Zichao Li [1], You Li [1,*] and Rongzheng Luo [2]

[1] School of Aerospace Science and Technology, Xidian University, Xi'an 710126, China; lizichao2022@163.com
[2] Institute of Remote Sensing Satellite, China Academy of Space Technology, Beijing 100094, China; lrz_519894574@163.com
* Correspondence: liyou@xidian.edu.cn; Tel.: +86-18392070397

Abstract: This paper improves the timeliness of satellite mission planning to cope with the rapid response to changes. In this paper, satellite mission planning is investigated. Firstly, the satellite dynamics model and mission planning model are established, and an improved Monte Carlo tree (Improved-MCTS) algorithm is proposed, which utilizes the Monte Carlo tree search in combination with the state uncertainty network (State-UN) to reduce the time of exploring the nodes (At the MCTS selection stage, the exploration of nodes specifically refers to the algorithm needing to decide whether to choose nodes that have already been visited (exploitation) or nodes that have not been visited yet (exploration)). The results show that this algorithm performs better in terms of profit (in this paper, the observation task is given a weight of 0–1, and each planned task will receive a profit; that is, a profit will be assigned at the initial moment) and convergence speed compared to the ant colony algorithm (ACO) and the asynchronous advantage actor critic (A3C).

Keywords: Monte Carlo tree search; timeliness; autonomous mission planning

check for updates

Citation: Li, Z.; Li, Y.; Luo, R. Satellite Autonomous Mission Planning Based on Improved Monte Carlo Tree Search. *Symmetry* **2024**, *16*, 1039. https://doi.org/10.3390/sym16081039

Academic Editor: Christos Volos

Received: 24 May 2024
Revised: 22 July 2024
Accepted: 29 July 2024
Published: 13 August 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Observation satellites can realize the observation of ground targets and obtain target status information through scanning, information sensing, and other methods by carrying various types of onboard remote sensors. Thanks to the advantages of satellites with a wide observation range and no regard for terrain constraints, satellites have an important role to play in the field of observation. The traditional satellite mission planning process is as follows: (1) The user puts forward the requirements. (2) The ground solves the observation sequence. (3) The command is sent to the imaging satellite. This completed process has enabled the effective control of imaging satellites. Satellite mission planning based on the ground centralized control mode has been generally accepted by various countries. However, with the gradual complexity of user mission requirements, the gradual increase in the number of ground target observation points, and the gradual increase in the number of satellites, the traditional control mode has been unable to meet the multifaceted needs of users and the establishment of a new mission planning mode is imminent.

In the past few decades, scholars in related fields have shown great interest in the problem of satellites. Gregory Beaumentl et al. designed the on-planet stochastic iterative greedy algorithm to decide an action [1]. Wolfe established an integer planning model for satellite scheduling based on the backpack problem [2]. Lemaître et al., for agile satellites, compared four algorithms such as greedy strategy, dynamic planning, constrained planning, and local search [3,4]. Habet proposed a taboo search algorithm based on local enumeration, which was combined with the search to solve the mission planning problem of satellites [5]. Bistra Dikina et al. combined an ascending read-first search with constrained propagation to solve the problem of autonomous satellite mission planning, but the profit and convergence rate need to be improved [6]. Hao et al. have established a multi-objective agile satellite scheduling model that considers mission consumption, mission completion

rate, and application profit and propose a heterogeneous algorithm that combines the ant colony algorithm (ACO) and genetic algorithm (GA) [7]. There are many studies on solving satellite mission planning based on heuristic algorithms such as genetic algorithm, simulated annealing, and tabu search, but Wang [8] analyzed that this method faces the problem of "difficult modeling", and Huang [9] found that this method has shortcomings in the real-time processing of sudden emergency missions.

Liu Song et al. designed the model and framework of on-board autonomous mission planning and proposed a rolling programming heuristic algorithm [10]. Chu X G et al. established a mathematical model of agile mission planning and designed an on-board planning solution algorithm based on branch demarcation [11]. Miao Yue et al. designed a real-time planning algorithm that takes into account the solution speed and result profit [12]. She Y C et al. proposed a new on-board mission planning method based on an improved mixed integer linear algorithm [13]. Xin et al. combined machine learning with satellite mission planning based on the neural network decision-making model, but it needed to rely on a large number of datasets [14]. Wang et al. have proposed an online scheduling algorithm for satellite mission planning based on the Markov decision model, but the algorithmic profit is low compared with satellite mission planning in the traditional control mode [15].

At present, there are two ways to carry out satellite mission planning: One is to solve the calculation on the ground and send the result to the observation satellite so as to realize the planning of the target mission on the ground. When a new observation demand arises, the method based on ground-based centralized planning requires the current planning mission to be solved before the new observation demand can be completed again. However, when sudden events occur, this method cannot realize the rapid response to emergency missions. At the same time, with the continuous improvement in user needs, this approach will lead to a low solution rate, which cannot meet user needs. The other way is a new control mode that combines artificial intelligence with satellite mission planning, which uses neural networks to make decisions and reinforcement learning to find the optimal strategy in the past experience. Compared with ground solving to realize mission planning, this method can change the user's needs in real-time and has a very high timeliness, but the method has a low profit.

Both of the mission planning models proposed above have drawbacks and cannot meet the needs of timeliness and high returns at the same time. An improved Monte Carlo tree search (Improved-MCTS) algorithm is proposed in this paper that can improve the profit and satisfy the timeliness requirements.

- The orbital dynamics model and the constraint satisfaction model are established by considering the effect of J2 perturbation;
- Combining Monte Carlo tree search with state uncertainty networks for a novel satellite mission planning model;
- The algorithm is compared with two mission planning models.

2. Problem Modeling and Grounded Theory

2.1. Orbital Dynamics Model

In this paper, the Orbit elements are used to describe the orbit of a satellite. The Orbit elements is a description of the six parameters necessary to determine the orbit of a celestial body or spacecraft moving in its Keplerian orbit under the action of Newton's laws of motion and the laws of gravity, including Semi major axis a, Eccentricity e, Inclination i, Argument of periapsis ω, Right Ascension of the Ascending Node Ω, and True Anomaly f.

The position vector $\mathbf{r}$ in the earth centered inertial (ECI) is obtained from orbit elements.

$$\mathbf{r} = \frac{a(1-e^2)}{1+e\cos f}\begin{bmatrix}\cos\Omega\cos(\omega+f)-\sin\Omega\sin(\omega+f)\cos i\\ \sin\Omega\cos(\omega+f)-\cos\Omega\sin(\omega+f)\cos i\\ \sin(\omega+f)\sin i\end{bmatrix}. \quad (1)$$

A derivation from the above equation yields velocity vector **v**.

$$\begin{aligned}\mathbf{v} &= \tfrac{d\mathbf{r}}{df}\cdot\tfrac{df}{dt}\\ &= \sqrt{\tfrac{\mu}{a(1-e^2)}}\left\{ e\sin f\begin{bmatrix}\cos\Omega\cos(\omega+f)-\sin\Omega\sin(\omega+f)\cos i\\ \sin\Omega\cos(\omega+f)+\cos\Omega\sin(\omega+f)\cos i\\ \sin(w+f)\sin i\end{bmatrix}\right.\\ &\left.+(1+e\cos f)\begin{bmatrix}-\cos\Omega\cos(\omega+f)-\sin\Omega\sin(\omega+f)\cos i\\ -\sin\Omega\cos(\omega+f)+\cos\Omega\sin(\omega+f)\cos i\\ \cos(\omega+f)\sin i\end{bmatrix}\right\},\end{aligned} \tag{2}$$

where $\mu = 398600.4415\ \mathrm{km}^3/s^2$ is the gravitational constant.

The conversion formula for the ECI coordinate system position velocity vector to orbit elements is as follows [16].

$$\begin{cases} a = \frac{p}{1-e^2} = \frac{h^2}{\mu(1-e^2)},\\ e = \frac{1}{\mu}[(v^2-\frac{\mu}{\mathbf{r}})r-(\boldsymbol{r}\cdot\boldsymbol{v})\boldsymbol{v}],\\ \cos i = \frac{\boldsymbol{k}\cdot\boldsymbol{h}}{h},\\ \cos\omega = \frac{\boldsymbol{n}\cdot\boldsymbol{e}}{ne},\\ \cos\Omega = \frac{\boldsymbol{i}\cdot\boldsymbol{n}}{n},\\ \cos f = \frac{\boldsymbol{e}\cdot\boldsymbol{r}}{er}, \end{cases} \tag{3}$$

where $\boldsymbol{h} = \boldsymbol{r}\times\boldsymbol{v}$ is the orbital angular momentum, $p = \frac{h^2}{\mu}$ is semi latus rectum, $\boldsymbol{i},\boldsymbol{j},\boldsymbol{k}$ is the triaxial unit vector, and $\boldsymbol{n} = \boldsymbol{k}\times\boldsymbol{h}$ is the ascending intersection vector.

2.2. Orbital Propagator

Considering that mission planning requires high accuracy for observing satellites, it is necessary to consider the influence of the Earth's J2 perturbation [17]. However, using the J2 model to forecast the orbit will generate a large amount of computation, resulting in a long computation time, which makes it difficult to meet the timeliness characteristics of mission planning. Therefore, this paper adopts an orbit model that considers the influence of the J2 long-term term and uses the orbit elements to forecast the orbit.

In a short period of time, the Semi major axis a, Eccentricity e, and Inclination i can be approximated to be unchanged, while the Argument of periapsis ω, Right Ascension of the Ascending Node Ω, and Mean Anomaly M change with time at the rate of [18]

$$\begin{cases} \dot{\omega}_{J_2} = C_{J_2}(3-\frac{5}{2}\sin^2 i)/(1-e^2)^2,\\ \dot{\Omega}_{J_2} = -C_{J_2}\cos i/(1-e^2)^2,\\ \dot{M}_{J_2} = C_{J_2}(1-\frac{3}{2}\sin^2 i)/(1-e^2)^2, \end{cases} \tag{4}$$

$$C_{J_2} = 1.5J_2R_e^2\sqrt{\mu}\bar{a}^{7/2}, \tag{5}$$

where $J_2 = 1.08264\times10^{-3}$ is the J_2 perturbation constant, $\mu = 398600.4415\ \mathrm{km}^3/\mathrm{s}^2$ is the gravitational constant, and $R_e = 6378.14$ km is the average radius of lateness. $\bar{a}$ denotes the mean half-length axis, and the transformation from the close half-length axis a to the mean half-length axis is

$$\bar{a} = a\left\{1-\frac{J_4R_e^2}{2a^2}\left[(2\cos^2 i-1)(\frac{a^3}{r^3}-\frac{1}{(1-e^2)^{3/2}})+3(1-\cos^2 i)(\frac{a}{r})^3\cos(2\omega+2f)\right]\right\}. \tag{6}$$

After time t, the Orbital elements become

$$\begin{cases} \omega_t = \omega+\dot{\omega}_{J_2}t,\\ \Omega_t = \Omega+\dot{\Omega}_{J_2}t,\\ M_t = M+(\sqrt{\mu/\bar{a}^3}+\dot{M}_{J_2})t. \end{cases} \tag{7}$$

The angle E of the declination point can be found from the angle of the flat declination point

$$M = E - e \sin E. \tag{8}$$

Which in turn yields the true proximal point angle f [19]

$$E = 2\arctan(\sqrt{\frac{1-e}{1+e}} \tan \frac{f}{2}). \tag{9}$$

3. Mission Planning for Observation Satellites Based on Improved-MCTS

3.1. Markov Decision Process for Mission Planning

A Markov Decision Process (MDP) is a theoretical framework for goal achievement through interactive learning. An agent observes the current state of the environment and then selects an action. The environment receives the action and transfers it to a new state and gives feedback, usually an immediate reward, based on the current state and the executed action [20]. As shown in Figure 1, at any moment, t, the agent chooses an action, A_t under the condition of the environment state, S_t. At the next moment, $t+1$, the agent receives the reward, R_t, of the action and updates the environment state it is in S_{t+1}.

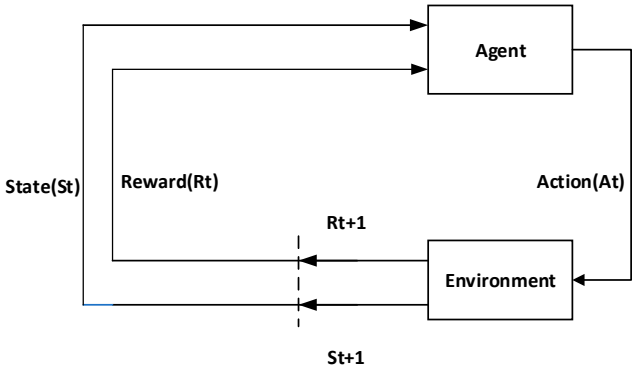


Figure 1. Agent interacting with the environment.

The probability that the next state of an agent is S_t depends only on the previous state S_{t-1} and the previous action A_{t-1}, independent of earlier states and actions, and is said to be Markovian in nature. If the previous state s and action a are given, the probability that the next state of the agent is s' and profit r at the moment is

$$p(s', r \mid s, a) = \Pr\{S_t = s', R_t = r \mid S_{t-1} = s, A_{t-1} = a\}. \tag{10}$$

For all $s \in S$,$a \in A(s)$, satisfy

$$\sum_{s' \in S} \sum_{r \in R} p(s', r \mid s, a) = 1. \tag{11}$$

The probability of an action with a selection at moment t and with the state s condition is

$$\pi(a \mid s) = \Pr\{A_t = a \mid S_t = s\}. \tag{12}$$

Under the condition of state s, action a, the value of the action is

$$Q(s, a) = \Pr\{A_t = a \mid S_t = s\}. \tag{13}$$

In satellite mission planning, the future state of the environment is only related to the current decisions and the current state of the satellite and not directly related to what decisions have been made in the past. Therefore, the satellite mission planning problem is Markovian in nature. In this paper, the satellite mission planning problem is described as a

Markov Decision Process (MDP). The mapping relationship from the environment state to the decision result is regarded as a decision strategy in MDP.

3.2. Modeling of Mission Planning for Observation Satellites

3.2.1. Time Window Selection

The time when a ground target is within the observable range of a satellite is defined as the visible time window (VTW). Unlike non-agile satellites that can only observe the target through the side swing angle, agile satellites use a pitch axis to extend the VTW. Assuming that the maneuvering times of agile satellites in roll, pitch and yaw directions are Δt_{φ}, Δt_{θ}, Δt_{ψ}, the maneuvering time of the satellite is

$$\Delta t_{man} = \max\{\Delta t_{\varphi}, \Delta t_{\theta}, \Delta t_{\psi}\}. \tag{14}$$

Considering the imaging time, Δt_{img}, the mission transition time is

$$\Delta t_{trans(j,j+1)} = \Delta t_{img} + \Delta t_{man(j,j+1)}. \tag{15}$$

If the observation moment is satisfied,

$$t_{j+1} \geq t_j + \Delta_{\mathrm{trans}(j,j+1)}, t_{j+1} \in VTW_{j+1}, \tag{16}$$

where t_{j+1} is the moment to perform the $j+1th$ mission, and t_j is moment to perform the jth mission. Then, the target is considered observable.

3.2.2. Mission Uniqueness

Each observation target can only be executed once, i.e., for $\forall \iota \in 1, \ldots\ldots, n$, there is

$$Y(i,t) = \begin{cases} 1, ith \text{ target observed}, \\ 0. \end{cases} \tag{17}$$

A satellite can only observe one target at the same moment.

3.2.3. Action Collection

Setting the action

$$a = \begin{cases} 1, \ if \ accept, \\ -1, \ else. \end{cases} \tag{18}$$

3.2.4. State Space

The transition time between neighboring observation missions depends on the attitude angle of the satellite performing the two observation missions. The attitude angle depends on the positions of the satellite and the target at the moment of observation. When the initial value of the satellite orbit as well as the coordinates of the two target points i and $i+1$ are given, the transition time of the missions is only related to the moment of the beginning of the observation, i.e.,

$$\Delta t_{trans(i,i+1)} = g(t_i, t_{i+1}).$$

Defining st_i and et_i as the start and end moments of observing the first i mission, then

$$et_i + \Delta t_{trans(i,i+1)} < st_{i+1}. \tag{19}$$

Subject to the limitations of satellite storage capacity, the data generated by the satellite observation target cannot exceed the maximum storage capacity of the satellite. For any satellite, the following formula exists:

$$\sum_i Stor_i \leq Stor_{\max}. \tag{20}$$

We define ω_i as the observation profit of the *ith* satellite.

We define $\sum_{i=1}^{max} p_i^{conflict}$ as the conflict loss of observing the *ith* target. The larger this value is, the higher the conflict between the current mission and the subsequent mission, which ultimately leads to a lower total profit. As shown in Figure 2, the shaded part is the portion of the *ith* target that is in conflict with the subsequent targets. The conflict loss value is the sum of the profit of the shaded target points.

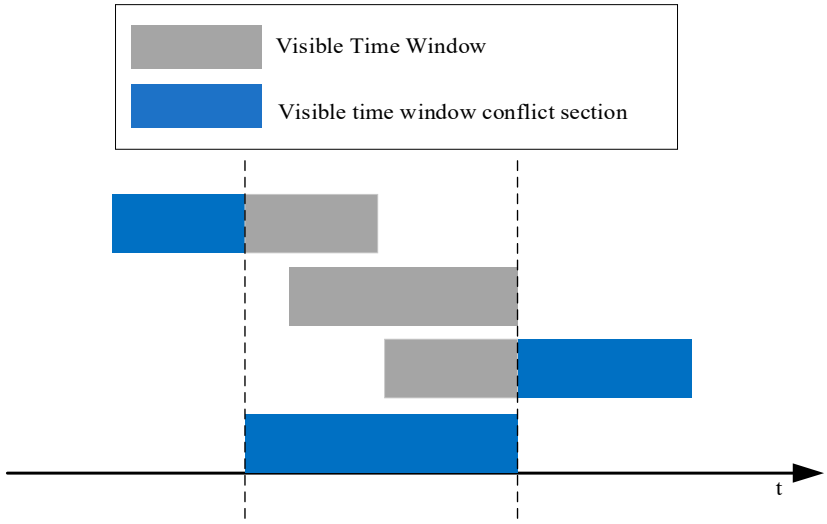


Figure 2. Conflict loss.

3.3. Satellite Mission Planning Based on Monte Carlo Tree Search

The strategy for the Monte Carlo tree search is divided into two phases: The first one is the in-tree strategy, which is mainly responsible for choosing the next action to be taken. The second one is the default strategy, which is needed to simulate the completion of the sampling of the whole sequence of states if there is no node containing the current state in the search tree. The default policy can use either a randomized policy or a policy based on the objective value function. By combining in-tree and default strategies, the Monte Carlo tree search is able to use simulated sampling to evaluate the value of each state and continuously expand the search tree [21]. Eventually, we can find the optimal action strategy. Figure 3 summarizes several major steps of the Monte Carlo tree search algorithm.

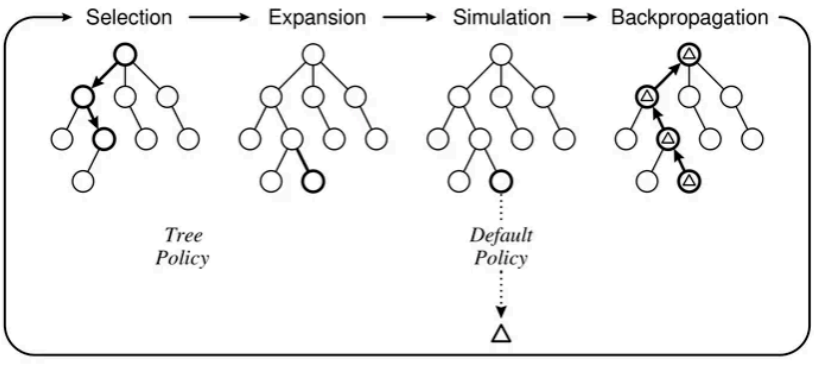


Figure 3. Schematic diagram of the Monte Carlo tree search algorithm.

In satellite mission planning, the state space has symmetry. For example, a picture of a target imaged by a satellite has rotational and mirror symmetries. Using these symmetries, symmetry-equivalent states can be regarded as the same state, which reduces the size of the state space, significantly reduces the number of branches in the search tree, and improves the search efficiency. At the same time, symmetry can help MCTS to carry out more effective pruning. In the search process, if the symmetry of a state has been explored, the previous results can be directly used to avoid repeated calculations, which greatly saves computational resources.

1. Selection: starting from the root node, the optimal child nodes are selected until a node most worthy of expansion is reached, i.e., the node represents a non-terminal state of the game and has not yet been expanded;
2. Expansion: if the node is not a termination node, then according to the currently available actions to create one or more child nodes, we should choose one;
3. Simulation: starting from this node, a simulation is run based on the default strategy until the end of the game.
4. Backpropagation: we use the results of the simulation to update the current node and its parent node.

The main goal of MCTS in satellite mission planning is to select the best next state s' given the current satellite state s. Figure 4 shows the selection phase of MCTS in satellite planning. The current state of the satellite is the root node, which can observe three target points, and for target point 1, the satellite chooses an action (observe or not) until it reaches an unexpanded node. The selection strategy controls the balance between exploitation and exploration, and the selected child node represents the optimal choice for the current state of the environment.

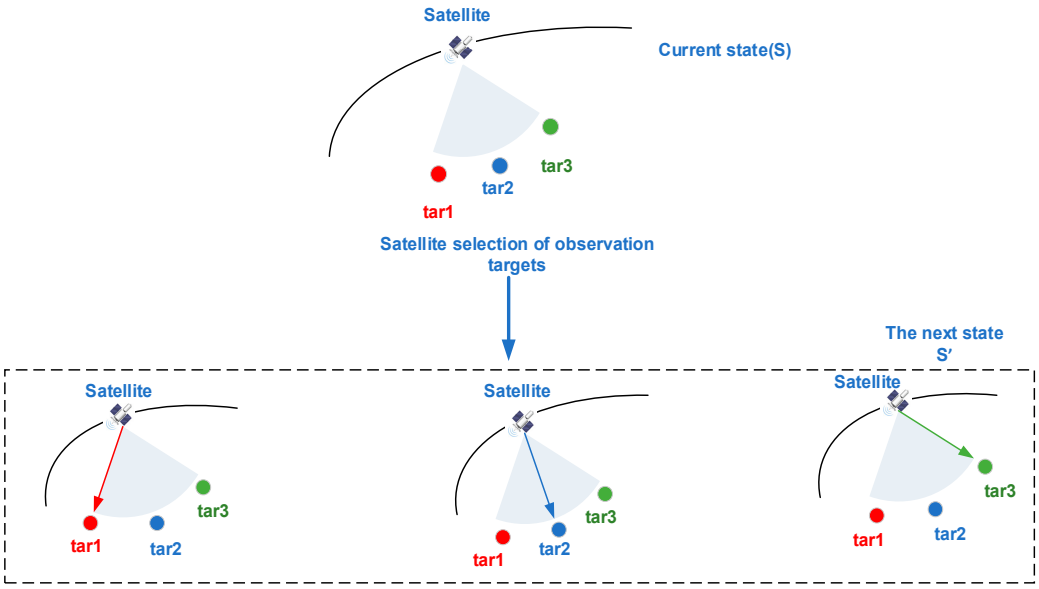


Figure 4. MCTS selection.

The expansion phase of MCTS is shown in Figure 5. For most domains, the entire game tree cannot be stored in memory, and the expansion determines whether for a given state as the root node, the node will expand by storing one or more child node in memory. In satellite planning, a node is added every time a satellite observes a target point. When mission planning is not completely finished, one or more child node can be created based on actions that are optional for the current state.

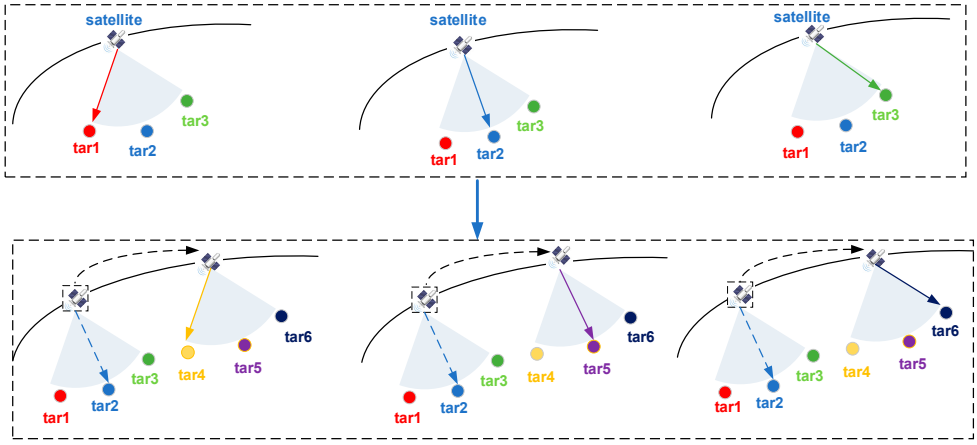


Figure 5. MCTS expansion.

Figure 6 shows the simulation phase of MCTS. The simulation is an MCTS process that selects the steps to move until the end of satellite mission planning. In this phase, default strategies are usually used, and the use of a proper simulation strategy can improve the final profit.

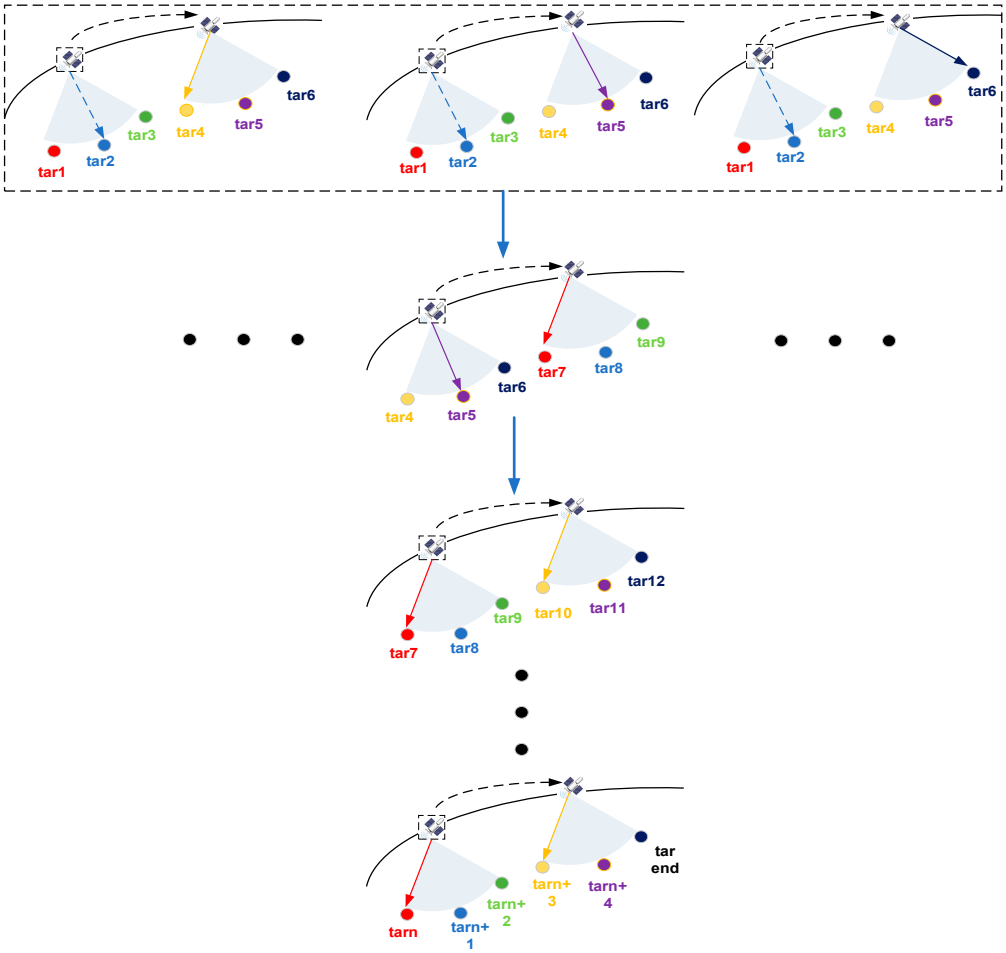


Figure 6. MCTS simulation.

Figure 7 shows MCTS backpropagation. Simulating the profits from satellite mission planning to update the current node as well as path-related root nodes. During the mission planning process, if the profit is higher than before, then $r_i(s_i, a_i) = +1$, and vice versa $r_i(s_i, a_i) = -1$. At time i, under the condition of state s, $r_i(s_i, a_i) = +1$ denotes the profit performing action a. If it is higher than the previous simulation, the profit is +1, and vice versa. The most efficient strategy for calculating the value of a node's action using backpropagation is to find the average of the results obtained by this node through all the simulations.

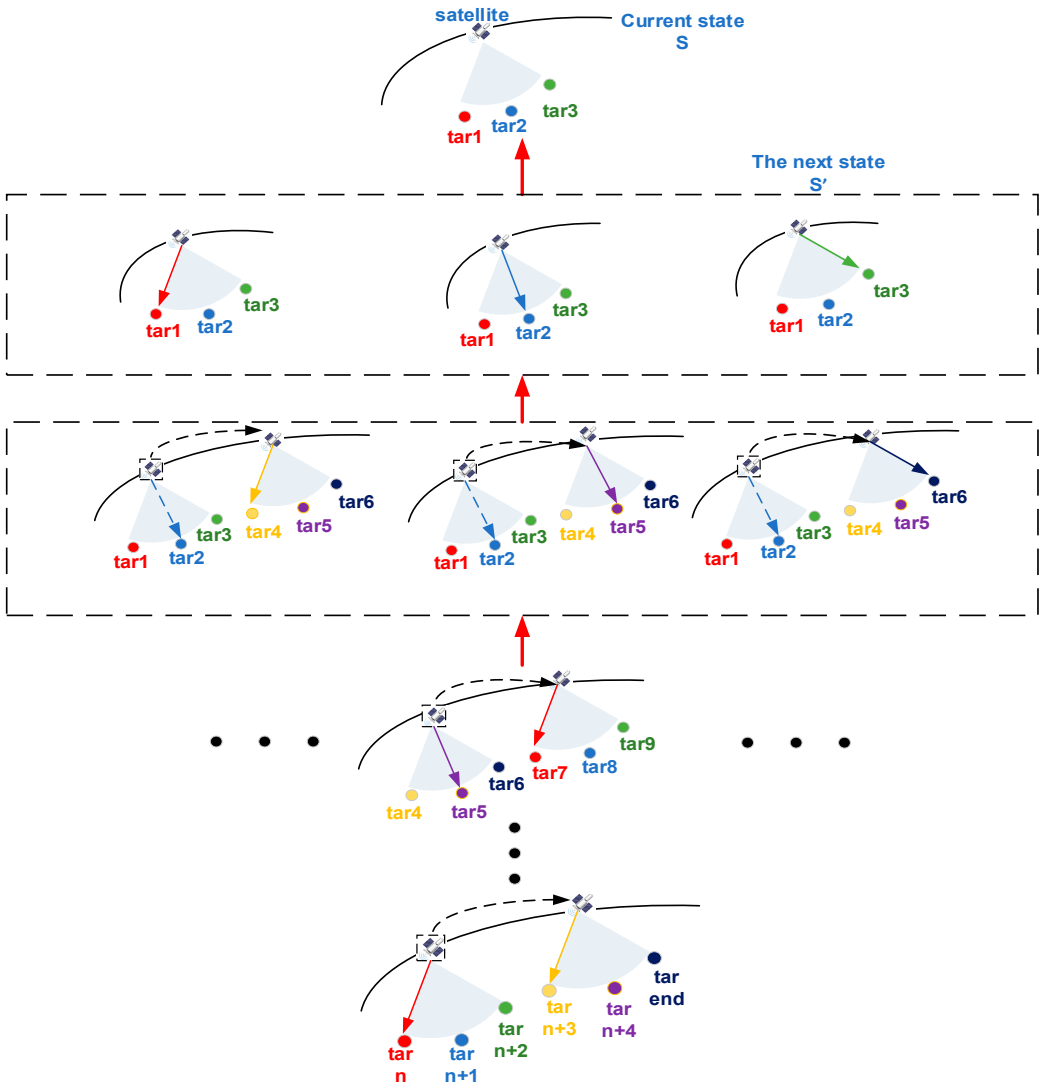


Figure 7. MCTS backpropagation.

3.4. Satellite Mission Planning Based on Improved-MCTS

This algorithm includes MCTS, a state uncertainty network (State-UN), and an Improved-MCTS. The overall framework is shown in Figure 8.

During the mission planning process, the current state data is collected as training data for the state uncertainty network. The output of the network is the uncertainty u, the action strategy p, and the probability of higher profit than before v, and the output of the state uncertainty network is involved in the improved Monte Carlo tree search, which ultimately trains a model for satellite mission planning.

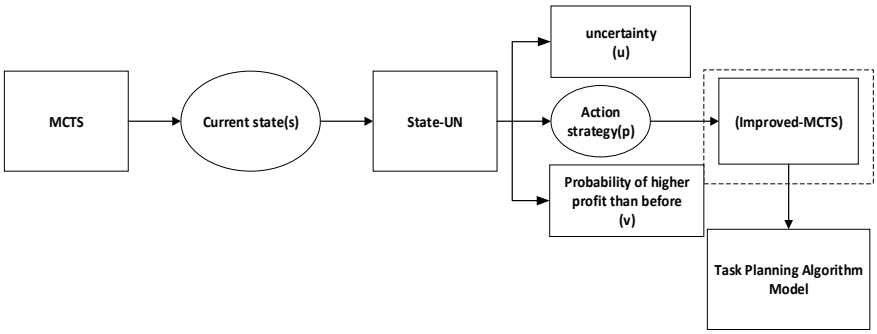


Figure 8. Satellite mission planning structure.

3.4.1. Uncertainty Assessment for MCTS

When the Monte Carlo tree search performs simulations with more data, it requires longer computation times, which can lead to very slow convergence [22]. However, not every state requires a long search time to determine the optimal action. Therefore, this section proposes an uncertainty estimation for MCTS, which reduces the simulation of satellite search states during MCTS and speeds up the training of the satellite mission planning model by predicting the uncertainty of the current state and stopping the search as soon as a stable optimal action is found.

Uncertainty estimation can be used for better exploration as well as avoiding risky and unknown choices. The determinism of the MCTS state is first defined. As shown in Figure 9, target 1 is the only point target to be considered by the satellite, and in this state, the root state is evaluated, actions are determined and the search is stopped immediately. Figure 10 represents the state with multiple action choices, where the agent is unable to determine which target point is best to choose after a small number of searches.

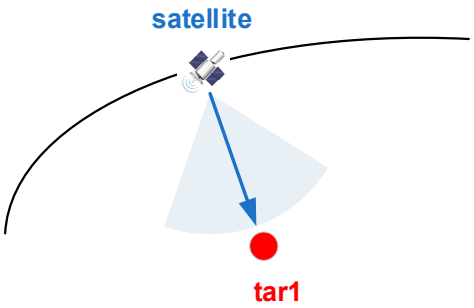


Figure 9. Satellite mission planning action selection under certainty.

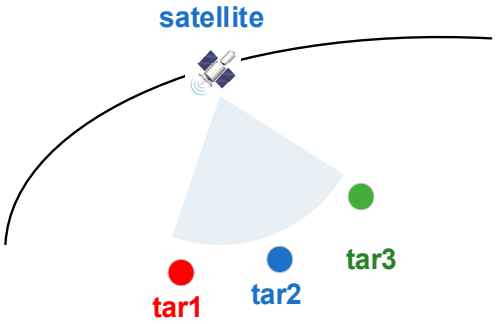


Figure 10. Satellite mission planning action selection under uncertainty.

In this paper, we define the uncertainty of the current search state based on the following observation.

1. By combining neural networks and MCTS after a maximum number of simulations, the observation strategy and valuation are used to output the true value of the simulated mission. During the planning process, Improved-MCTS converges to a high profit after hundreds of simulations. When a sufficiently large maximum number of simulations is passed, it is easier to predict the uncertainty of the current search state.
2. In satellite mission planning, there may be a choice of multiple targets at the same time, but only the best target needs to be observed. If the agent in mission planning is certain that the target is the best, then the state is considered to be certainty. The best target is defined as the target with the highest action value after a maximum number of simulations.
3. Even though the agent can predict the best action strategy at the beginning of the MCTS, the state of the search is uncertain if another suboptimal target is considered the best observed target during the search period due to the fact that luck occupies an important factor at the beginning of the tree search.

Based on the above observations, a definition of uncertainty is proposed in this paper. Since it is necessary to consider ending the MCTS early and recording the tree state information of the MCTS after n iteration of simulations for predicting the uncertainty of the Monte Carlo tree search. So, let $\pi(a \mid s,n)$, $Q(s,a,n)$ be the action strategy and action value provided after n iteration of simulations for the satellite mission planning state s.

In this paper, we use the parameter τ to control the exploration level of MCTS, and $\tau = 1$ is the initial value. In the initial phase, the parameter τ encourages MCTS to explore in the self-play phase; the value of the parameter τ is gradually reduced to zero as the mission progresses.

$$\pi(a \mid s) = \frac{N(s,a)^{\frac{1}{\tau}}}{\sum_b N(s,b)^{\frac{1}{\tau}}}, \tag{21}$$

where $N(s,a)$ denotes the number of visits to select mission a in state s. $\sum_b N(s,b)$ denotes the sum of the number of visits to mission b in state s.

The action strategy π is selected using $\tau \to 0^+$. If a target has not been observed, the action value $Q(s,a',n)$ will be -1, indicating failure. Given the best simulation $N_{\max}$, the action value $Q(s,a,N_{\max})$ is used to approximate the true action value. Thus, the approximate reward function R for the current target observation strategy can be obtained as

$$R_{N_{\max}}(s,n) = \sum_a (\pi(a \mid s,n) * Q(s,a,N_{\max})), \tag{22}$$

where $\pi(a \mid s,n)$ denotes the probability of choosing mission a when state s and number of simulations n. $Q(s,a,N_{\max})$ denotes the action value of choosing action a in state s under the maximum number of simulations. The formula calculates the overall profit by multiplying the cumulative strategy probability with the action value.

Define MCTS after n simulation as $U(s,n) = 1$, if

$$\exists n' \geq n (R_{N_{\max}}(s,N_{\max}) - R_{N_{\max}}(s,n') \geq \varepsilon), \tag{23}$$

where ε is a parameter between 0 and 1. The tree state, $U(s,n) = 0$, can be determined when each $n' \geq n$ selects a near-optimal observation action. When the environment state is s, the minimum simulation count, $M(s)$, is defined as the search state becomes determined after $M(s)$ simulations. That is, if $n < M(s)$, $U(s,n) = 1$, otherwise $M(s) \leq n \leq N_{\max}$ and $U(s,n) = 0$. As shown in Figure 8, the minimum simulation count is $M(s) = 1$.

This is carried out by providing the value $u \in [-1,+1]$ such that the model in this paper predicts the probability of $U(s,n) = 1$. If u is large, $U(s,n)$ is more likely to be equal to 1. If u is smaller than an adjusted threshold, the search can be stopped.

3.4.2. MCTS Uncertainty Prediction

The state uncertainty network (State-UN) is used to predict the current planning uncertainty. The uncertainty network takes as input only the current planning state s (i.e., profit, current satellite storage capacity, remaining storable capacity, and observation start moment), and the outputs are the uncertainty value u, the policy p, and the probability v. The formula is as follows:

$$StateUN(s) = (u, p, v). \tag{24}$$

The state uncertainty network is used to predict the initial uncertainty defined in Equation (24) $U(s, 1)$, the Monte Carlo tree search strategy $\pi(* \mid s, N_{\max})$ after maximal simulation counts $N_{\max}$, and outcome Z of the mission planning. The loss function can be obtained as follows:

$$loss = (u - U(s, 1))^2 + c_1((v - Z)^2 - \pi^T \log(p)) + c_2\|\theta\|^2. \tag{25}$$

where c_1, c_2 is a weighting constant. The first term of the loss function uses the mean square error to make the value u close to $U(s, 1)$. The second term of the loss function, controlled by c_1, π and p are both vectors. $\pi^T \log(p)$ is their cross-entropy, which is added to help the network extract more state information. The last term is the L2 regularization, θ is a parameter of the algorithm model in this paper.

3.4.3. Satellite Mission Planning Based on Improved-MCTS

Since not all mission planning states require the Monte Carlo tree to run through the set number of simulations to obtain the optimal strategy, this paper proposes the Improved-MCTS algorithm to speed up the training of the satellite mission planning algorithm model by reducing the number of simulations of the Monte Carlo tree search.

The Improved-MCTS is based on the uncertainty u prediction of the State-UN about the current tree state. The action decides whether to continue with the simulation phase of the Monte Carlo tree search. If the current action is a stable action, then the search is immediately terminated and proceeds to the next phase; if it is not the optimal operation, then the simulation continues, and the structure of the algorithm is shown in Figure 11.

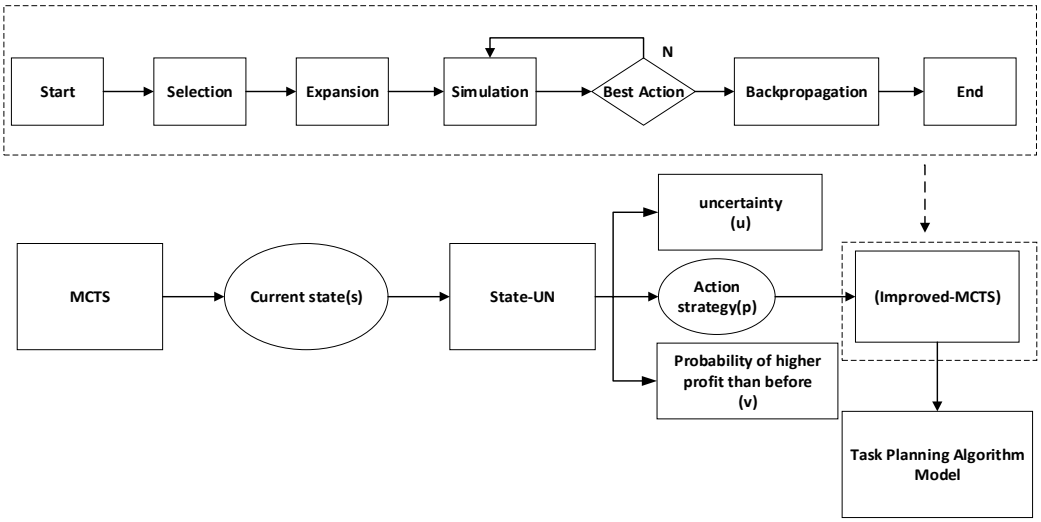


Figure 11. Algorithm structure.

Below is a flowchart of the entire algorithm (Figure 12):

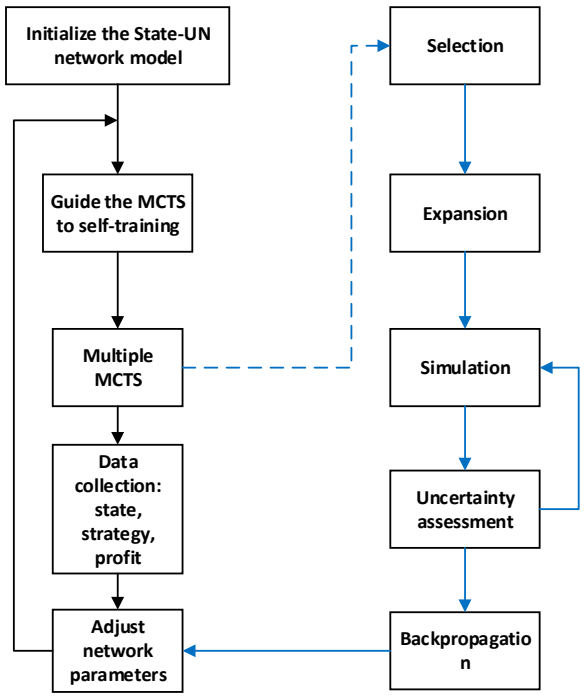


Figure 12. Flowchart of Improved-MCTS algorithm.

As shown in Figure 12, the steps are as follows:

1. Initialize:

At the beginning of the algorithm, a State-UN network model is initialized. The role of this model is to assess the state and provide a strategy.

2. Self-training:

We use the current State-UN to guide the MCTS for self-training. In each task planning, the algorithm performs multiple MCTS simulations to explore the most profitable missions.

At each step of planning, MCTS uses the evaluation of the State-UN to improve its search process. Network evaluation helps the algorithm decide which nodes to scale to in a tree search.

There is an uncertainty assessment during the simulation phase and the backpropagation phase. This is the core innovation of the algorithm. After each simulation, the State-UN is used to assess the uncertainty of the current node. If the uncertainty of the current node is low (i.e., the network predicts that the current move is close to optimal), further searches and simulations are terminated. This process reduces the number of simulations and improves the efficiency of MCTS.

3. Data collection:

In the process of self-training, the algorithm collects states, strategies, and profit.

4. Network training:

We use the collected data to train the State-UN. This training process adjusts the weights of the network through backpropagation.

5. Iterative optimization:

After a certain period of self-training, the performance of the State-UN will gradually improve. This improved network was once again being used to bootstrap the MCTS for more self-training.

4. Simulation Results and Analysis

4.1. Analysis of Training Results

To support our idea, we trained the State-UN and conducted some simulations. Parameters of the satellite and missions in these simulations are given in Table 1.

Table 1. Simulation parameters.

Parameter	Value
Mission counts	100
Training episodes	2000
Profit of mission ω_i	0.5–1
Storage consumption $Stor_i$	5 Gb
Max storage $Stor_{\max}$	200 Gb
Imaging time $\Delta timg$	5 s

In this paper, we use the total number of accepted missions and the total profit to evaluate the performance of the mission planning algorithm. The higher the mission counts, the greater the total profit of mission planning is likely to be. The more training episodes, the more accurate the results may be, but it will result in a long training time. As the number of training increases, the satellite will choose a mission with a larger ω_i for observation. Each mission consumes 5 GB of storage capacity, and the maximum number of planned missions is 40. $\Delta timg$ will affect the transition time between the two missions. The results are shown in Figures 13 and 14.

Figure 13 represents the profit achieved by satellite mission planning after 500 episodes of training, where the blue lines represent the measured values and the orange lines represent the trending values. In the phase from 0 to the 100th episode, the profit obtained from observation is in the random state (this period is equivalent to the first-come-first-served strategy). As the number of training episodes increases, the algorithm further adjusts its decision strategy and begins to favor better profit. In the phase of 100–400 episodes, the profit is further increased. When the training reaches 400 or more episodes, the gains gradually converge and improves to 14.

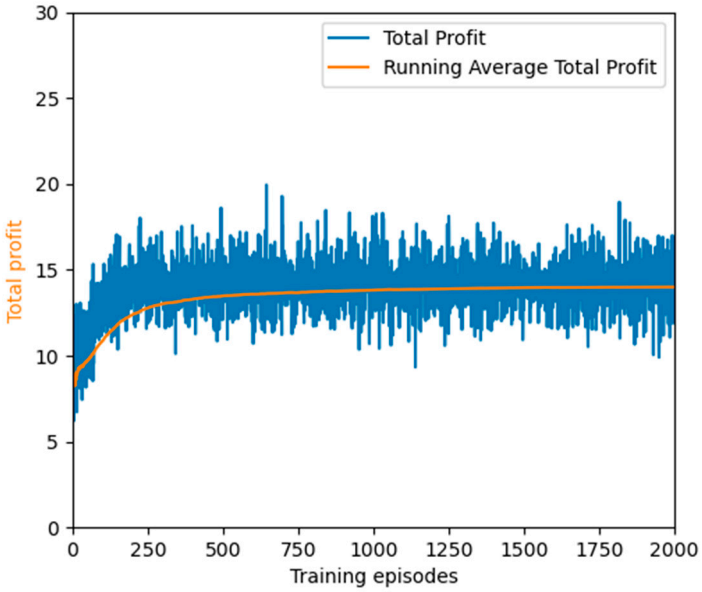


Figure 13. Total profit.

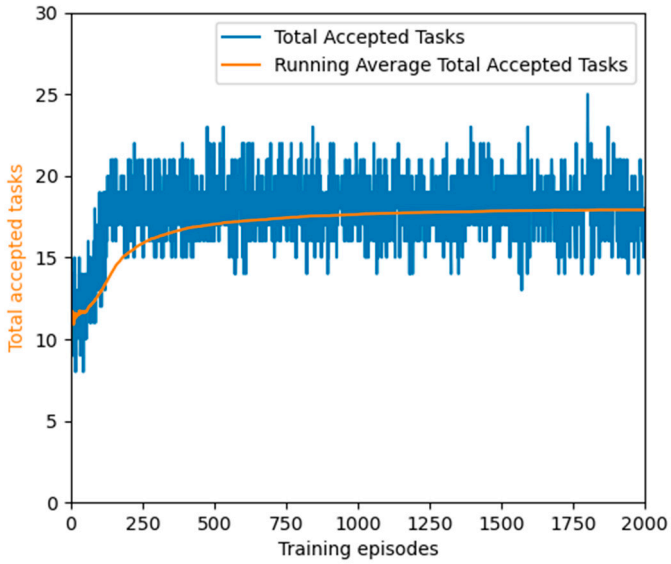


Figure 14. Total accepted missions.

Figure 14 represents the number of total missions accepted by the algorithm, where the blue line represents the measured value and the orange line represents the trend value. As the number of training times increases, the number of total missions accepted by the algorithm gradually improves in the phase. When the training reaches more than 500 episodes, the number of missions tends to converge and eventually improves to about 18.

4.2. Adaptability and Comparison

The main purpose of this section is to verify the generalization of the Improved-MCTS algorithm, not the overfitting strategy, for a specific dataset while comparing it with the ant colony (ACO) algorithm and asynchronous advantage actor critic (A3C) algorithm. The initial simulation scenario is determined as follows.

The mission start moment is 04:40:00 UTC on 22 June 2023. The mission planning time lasts 10 min, and the orbit elements of the satellite at the start moment are shown in Table 2.

Table 2. Orbit elements.

Orbit Elements	Value
Semi-Major Axis (km)	70,000.213
Eccentricity	0.00016
Inclination (rad)	0.5236
Right Ascension of the Ascending Node (rad)	0.3491
Argument of Periapsis (rad)	0.3432
True Anomaly (rad)	0.7040

In this paper, a number of target points are randomly generated in the track of the subsatellite point during the mission time interval, and the latitude and longitude of the target do not exceed 10° from the subsatellite point.

In Figure 15, the black line represents the track of the subsatellite point, the blue x represents the target point, and the orange line represents the target point that the satellite has received. The results of the mission planned using the algorithm in this paper have most of the targets distributed on the same side of the track of subsatellite point. This is because the agent is trained to reduce the maneuvering time of the satellite's attitude transition in order to obtain more observation opportunities for the targets. However, the ACO algorithm does not take this into account, resulting in fewer observation targets.

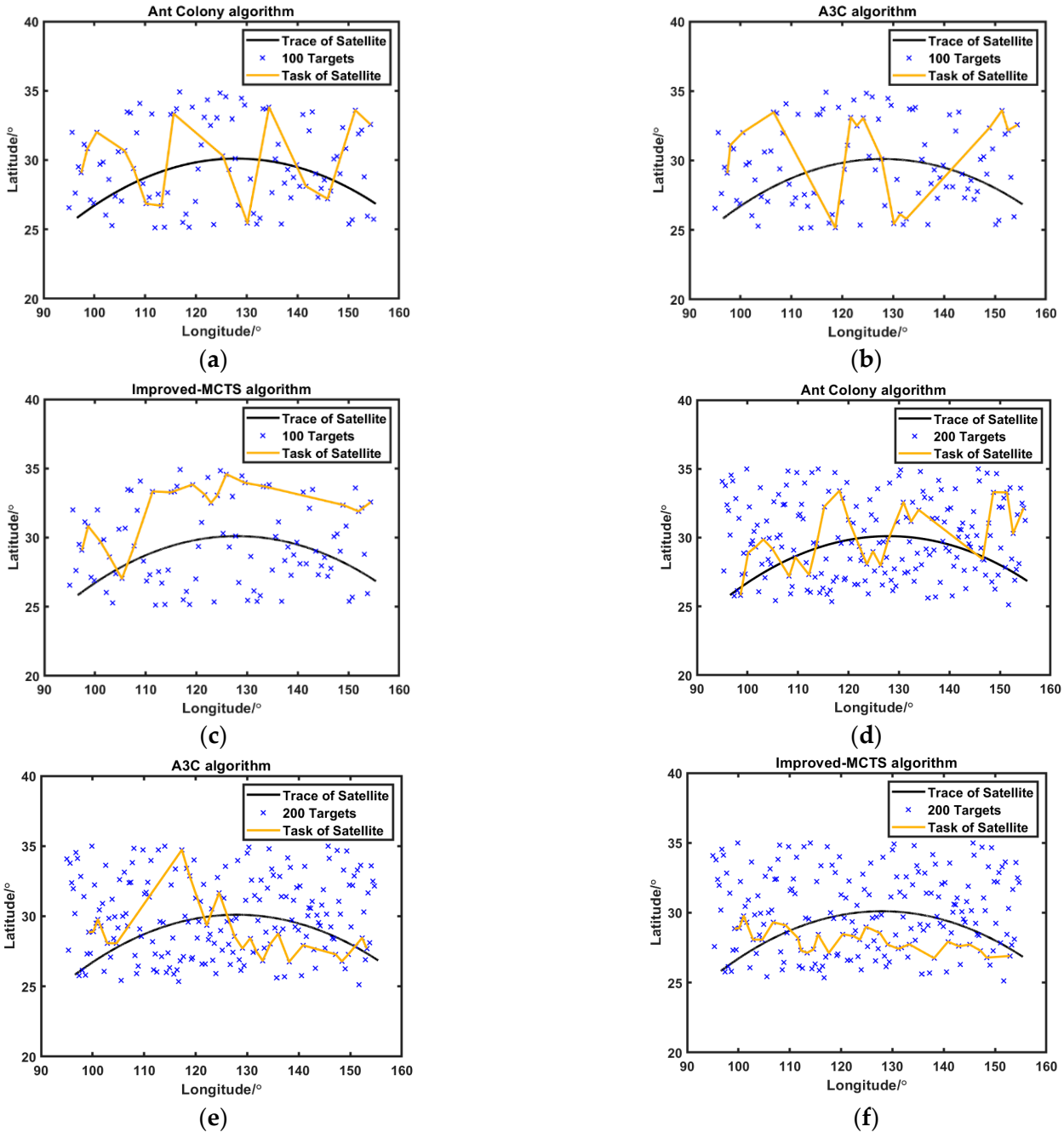


Figure 15. (**a**) Results of the 100 targets via ACO algorithm; (**b**) results of the 100 targets via A3C algorithm; (**c**) results of the 100 targets via Improved-MCTS; (**d**) results of the 200 targets via AC algorithm; (**e**) results of the 200 targets via A3C algorithm; (**f**) results of the 200 targets via Improved-MCTS.

We divided the experiments into six groups according to the mission count, and each group contained simulations of 10 different scenarios, and the results are shown in Figure 16. From the figure, we can visualize that the Improved-MCTS has higher profit than the other two algorithms, and the gap becomes more and more obvious as the mission count increases.

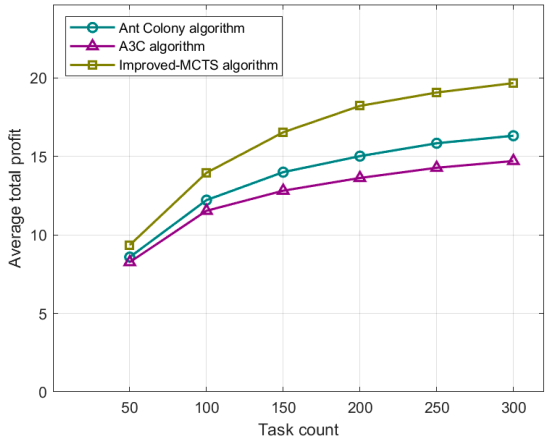
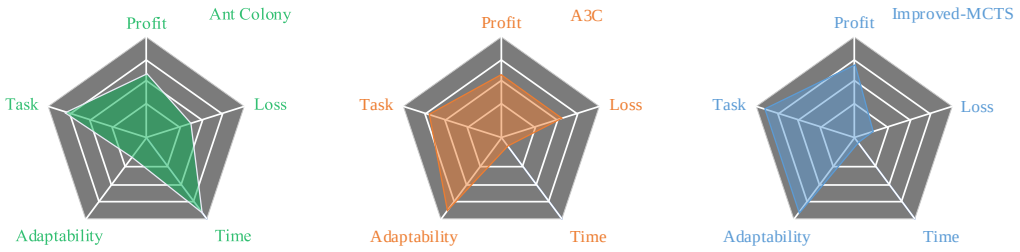


Figure 16. Comparison of the profit by the three algorithms.

We experimented with 20 different scenarios to compare the Improved-MCTS with the classical algorithm. The average total revenue, average number of accepted missions, and average mission response time are used as metrics. The simulation results are shown in Table 3 and Figure 17.

Table 3. Comparison (total mission count = 100).

	Ant Colony	A3C	Improved-MCTS
Average total profit	12.41	11.38	14.05
Average mission count	16.23	14.98	18.33
Average time	20.51	0.59	0.53
Average conflict loss	17.32	26.75	13.11
Adaptability	0	1	1
Response style	Batch	Immediate	Immediate



(**a**) (**b**) (**c**)

Figure 17. Performance comparison by the three algorithms. (**a**) Performance of the ACO algorithm; (**b**) performance of the A3C algorithm; (**c**) performance of the Improved-MCTS.

5. Discussion

Compared with the traditional satellite mission planning mode, the convergence speed of the proposed algorithm is much better than the ant colony algorithm. Compared with the new satellite mission planning mode, the profit of the proposed algorithm is much higher than the A3C algorithm. The agent trained by the algorithm is not specific to a particular scenario. When applied to new scenarios, the algorithm can quickly obtain the optimal planning results and meet the requirements of timeliness. Combining MCTS with an uncertainty network, the training processes affect each other. However, this paper only focuses on single-satellite mission planning and multi-satellite collaborative planning is required when a large number of targets appear. The collaborative planning of multi-satellite missions will be the focus of future research.

6. Conclusions

In order to meet the timeliness of task planning and improve the total profit of a mission, we propose a mission planning model based on an improved MCTS. In this paper, MCTS not only solves the problem of timeliness in satellite mission planning but also significantly improves the profit compared with ACO and A3C algorithms.

The State-UN can be trained for immediate response scheduling without the need for specialized expert knowledge. Our main contribution was proposing a real-time planning method for a satellite by importing MCTS and the State-UN into satellite mission planning. The experiment proves that the model not only meets the timeliness requirement but also greatly improves the total profit. In the future, we will extend the method to multi-satellite mission planning situations.

Author Contributions: Conceptualization, Y.L.; Methodology, Z.L.; Formal analysis, R.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Beaumet, G.; Verfaillie, G.; Charmeau, M.C. Autonomous planning for an agile earth-observing satellite. In Proceedings of the ISAIRAS, Los Angeles, CA, USA, February 2008.
2. Wolfe, W.J.; Sorensen, S.E. Three Scheduling Algorithms Applied to the Earth Observing Systems Domain. *Manag. Sci.* **2000**, *46*, 148–166. [CrossRef]
3. Lemaître, M.; Verfaillie, G. Daily management of an earth observation satellite. In Proceedings of the Comparison of ILOG International Users Meeting, Paris, France, July 1997.
4. Lemaître, M.; Verfaillie, G.; Jouhaud, F.; Lachiver, J.-M.; Bataille, N. Selecting and scheduling observations of agile satellites. *Aerosp. Sci. Technol.* **2002**, *6*, 367–369. [CrossRef]
5. Habet, D.; Vasquez, M. Saturated and Consistent Neighborhood for Selecting and Scheduling Photographs of Agile Earth Observing Satellite. In Proceedings of the Fifth Metaheuristics International Conference, Kyoto, Japan, 25–28 August 2003.
6. Dilkina, B.; Havens, B. *Agile Satellite Scheduling via Permutation Search with Constraint Propagation*; Actenum Corporation: Vancouver, BC, Canada, 2005.
7. Sun, K.; Yang, Z.Y.; Wang, P.; Chen, Y.W. Mission Planning and Action Planning for Agile Earth-Observing Satellite with Genetic Algorithm. *J. Harbin Inst. Technol. New Ser.* **2013**, *20*, 51–56.
8. Wang, C.; Jing, N.; Li, J. An algorithm of cooperative multiple satellites mission planning based on multi-agent reinforcement learning. *J. Natl. Univ. Def. Technol. China* **2002**, *33*, 53–58.
9. Huang, H.; Sun, C.Y.; Hu, J.X. Optimization design of response satellite deployment for regional target emergency observation. In Proceedings of the 2020 International Conference on Guidance on Advances in Guidance, Navigation and Control, Tianjin, China, 23–25 October 2020; Springer: Berlin/Heidelberg, Germany, 2022; pp. 579–591.
10. Liu, S.; Chen, Y.; Xing, L.; Sun, K. Method of agile imaging satellites autonomous task planning. *Comput. Integr. Manuf. Syst.* **2016**, *22*, 928–934.
11. Chu, X.G.; Chen, Y.N.; Tan, Y.J. An anytime branch and bound algorithm for agile earth observation satellite onboard scheduling. *Adv. Space Res.* **2017**, *60*, 2077–2090. [CrossRef]
12. Miao, Y.; Wang, F. Optimize-by-priority on-orbit task real-time planning for agile imaging satellite. *Opt. Precis. Eng.* **2018**, *26*, 150–160. [CrossRef]
13. She, Y.C.; Li, S.; Zhao, Y.B. Onboard mission planning for agile satellite using modified mixed-integer linear programming. *Aerosp. Sci. Technol.* **2018**, *72*, 204–216. [CrossRef]
14. Wang, X.; Wu, J.; Shi, Z.; Zhao, F.; Jin, Z. Deep reinforcement learning based autonomous mission planning method for high and low orbit multiple agile earth observing satellites. *Adv. Space Res.* **2022**, *70*, 3478–3493. [CrossRef]
15. Wang, H.; Yang, Z.; Zhou, W.; Li, D. Online scheduling of image satellites based on neural networks and deep reinforcement learning. *Chin. J. Aeronaut.* **2019**, *32*, 1011–1019. [CrossRef]
16. Zhang, R. *Satellite Orbital Attitude Dynamics and Control*; Beijing University of Aeronautics and Astronautics Press: Beijing, China, 1998; pp. 2–3.
17. Han, H.; Dang, Z. Models and Strategies for J2-Perturbed Orbital Pursuit Evasion Games. *Space Sci. Technol.* **2023**, *3*, 0063. [CrossRef]
18. Zhang, G.; Cao, X. Coplanar ground-track adjustment using time difference—ScienceDirect. *Aerosp. Sci. Technol.* **2016**, *48*, 21–27. [CrossRef]

19. Xiao, Y.; de Ruiter, A.; Ye, D.; Sun, Z. Attitude coordination control for flexible spacecraft formation flying with guaranteed performance bounds. *IEEE Trans. Aerosp Electron. Syst.* **2023**, *59*, 1534–1550. [CrossRef]
20. Jiang, R.; Ye, D.; Xiao, Y.; Sun, Z.; Zhang, Z. Orbital Interception Pursuit Strategy for Random Evasion Using Deep Reinforcement Learning. *Space Sci. Technol.* **2023**, *3*, 0086. [CrossRef]
21. Fu, M.C. Simulation-Based Algorithms for Markov Decision Processes: Monte Carlo Tree Search from AlphaGo to AlphaZero. *Asia Pac. J. Oper. Res.* **2019**, *36*, 1940009. [CrossRef]
22. Petschnigg, C.; Spitzner, M.; Weitzendorf, L.; Pilz, J. From a Point Cloud to a Simulation Model Bayesian Segmentation and Entropy Based Uncertainty Estimation for 3D Modelling. *Entropy* **2021**, *23*, 301. [CrossRef] [PubMed]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.