applied sciences

MDPI

Article

Toward Low Time Fragmentation of Equipment: A Double-DQN Based TT&C Task Planning Approach

Hangkun Xu and Runzi Liu *

School of Information and Control Engineering, Xi'an University of Architecture and Technology, Xi'an 710055, China
* Correspondence: rzliu@xauat.edu.cn

Abstract: With the increase of the number of satellites in space, satellite tracking, telemetry, and command (TT&C) is becoming more and more important for aerospace. This paper proposes a method for a low time fragmentation oriented. TT&C task planning method based on Double Deep Q-Network (DDQN). This method mainly solves the problem of poor responses to emergency tasks caused by the large amount of time fragments of equipment under traditional TT&C task-planning methods. Firstly, a multi-objective optimization model aiming at minimizing time fragments and maximizing task revenue is formulated. Then, according to the conflict characteristics of tasks, a task-planning conflict graph is proposed, based on which a TT&C task-planning problem is transferred into an independent set problem. Finally, DDQN is combined with graph structure embedding to solve the transferred independent set problem. The experimental results demonstrate that the proposed method reduces the time fragment of TT&C equipment by 32% and shortens the response time of emergency tasks by 36% compared to existing methods.

Keywords: deep reinforcement learning; low time fragmentation; satellite tracking, telemetry, and command; task planning

check for updates

Citation: Xu, H.; Liu, R. Toward Low Time Fragmentation of Equipment: A Double-DQN Based TT&C Task Planning Approach. *Appl. Sci.* **2023**, *13*, 6326. https://doi.org/10.3390/app13106326

Academic Editor: Paolo Renna

Received: 13 March 2023
Revised: 14 May 2023
Accepted: 18 May 2023
Published: 22 May 2023

Copyright: © 2023 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

The satellite tracking, telemetry, and command (TT&C) system is utilized to transmit the telemetry and telecommand data and determine the satellite orbit [1]. With the development and construction of prominent satellite constellations such as Hongyun and Hongyan in China [2], the number of satellites in space has rapidly increased [3,4], resulting in an increasing TT&C demand for space networks. On the other hand, due to the long construction period, high cost, and constraints of national borders, the number of ground stations is relatively limited, resulting in a lack of available TT&C resources. Therefore, more severe resource conflicts will be encountered in TT&C task planning, which also brings more significant challenges to the TT&C resource management of space networks.

The TT&C tasks can be divided into two categories: common tasks and emergency tasks. Common tasks [5] request the daily tracing of satellites in space and form a majority of the TT&C tasks. Emergency tasks [6] are the real-time tasks that take place during emergencies. As satellites play an increasingly important role in disaster relief and emergency search and rescue, more and more emergency TT&C tasks are emerging. During the planning stage, the emergency task often needs to be inserted into the idle time of the TT&C equipment or preempt the planned resources that are allocated to common tasks. However, the existing traditional TT&C task-planning methods only aim to maximize the sum revenue of the completed tasks, which generates many short time intervals (i.e., time fragments) between the TT&C windows, which cannot be used for planning tasks, significantly affecting the planning of emergency tasks. The resources of common TT&C tasks are preempted so that the common TT&C tasks cannot be successfully executed. Therefore, it is necessary to study the TT&C task-planning method toward low time

fragments, which facilitate the planning of emergency TT&C tasks to reduce the response time of emergency TT&C tasks.

The existing methods for common TT&C task planning mainly include deterministic algorithms [7–9], heuristic algorithms [10–12], and intelligent algorithms [13–17]. Among them, the deterministic algorithm is mainly suitable for solving TT&C task scheduling problems on a small scale, but it is difficult to apply this algorithm to solve problems on a large scale. The heuristic algorithm is mainly based on certain heuristic rules to obtain the optimal solution within a certain time. However, the heuristic rules of this type of algorithm are often derived from engineering experience and have great uncertainty. Intelligent algorithms seek optimal results while taking into account efficiency and accuracy. With the gradual increase in satellites in recent years, traditional deterministic and heuristic algorithms have not been the most suitable for solving TT&C task planning problems. Therefore, most of the current research focuses on the research into intelligent algorithms. References [13–15] all established mathematical models of TT&C resource planning models to minimize the workload and use different ant colony algorithms to solve them. Reference [16] designed an emergency task-planning algorithm based on a simulated annealing algorithm with the goal of maximizing the total task revenue. The algorithm reduces the probability of falling into local optimum through an iterative neighborhood search and instability strategy to realize the reasonable allocation of multi-dimensional TT&C resources. Reference [17] established a mathematical model of TT&C task planning with a minimized overall task satisfaction rate as the optimization goal. It proposed an improved DQN (Deep Q-Network)-based TT&C resource optimization configuration method to effectively solve the joint planning problem of heterogeneous TT&C network resources. This algorithm effectively improves the satisfaction rate of the TT&C tasks.

Although, References [13–17] used intelligent algorithms to improve the completion rate of TT&C tasks effectively, which do not control the time fragments of TT&C equipment during the task-planning process, resulting in untimely responses to emergency tasks. Therefore, this paper designs a low time fragmentation-oriented TT&C task-planning method (LTFTT&C) based on DDQN (Double Deep Q-network) to minimize time fragmentation and maximize task revenue. The algorithm uses deep reinforcement learning to mine the distribution rules of TT&C task resources of the space network. It constructs a common TT&C task feasible scheme that considers the task completion rate and emergency task response speed. Unlike other traditional TT&C task-planning algorithms, LTFTT&C can reduce the time fragments in the task-planning process while ensuring the success rate of the task. The basic idea is as follows: Firstly, aiming at the problem of high time fragmentation and low utilization rates of TT&C equipment in the existing TT&C task-planning results, a mathematical model of TT&C task planning is developed with the objective function of minimizing the total time fragments and maximizing the total task revenue. Then, based on the conflict graph model, the task-planning problem is transformed into a maximum independent set problem. Finally, the task-planning algorithm based on DDQN and the mathematical model are used alternately to complete the entire solution process. Comparing the LTFTT&C algorithm with the existing methods, the simulation results show that the proposed algorithm in this paper reduces the time fragments of TT&C equipment while ensuring the revenue of common TT&C tasks and effectively improves the response speed to subsequent emergency tasks.

The remainder of this paper is organized as follows: Section 2 gives a description of the TT&C task-planning scenario and problem. Section 3 has established the mathematical model of the TT&C task-planning problem in a space network. Section 4 transforms the TT&C task-planning problem into an independent set problem on the graph. Section 5 introduces a low time fragmentation-oriented TT&C task-planning algorithm. Section 6 conducts simulation experiments to verify the effectiveness of the proposed algorithm. The last section gives the research conclusion.

2. Scenario and Problem Description

The TT&C scenario shown in Figure 1 is composed of satellites and ground stations. The set of satellites is $S = \{s_1, s_2, s_3, \ldots, s_i, \ldots, s_N\}$, and s_i represents the i-th satellite. The set of ground stations is $G = \{g_1, g_2, g_3, \ldots, g_h, \ldots, g_M\}$, and g_h represents the h-th ground station. Each ground station has L TT&C equipment. The set of all the TT&C equipment is $A = \{a_1, a_2, a_3, \ldots, a_j, \ldots, a_L, a_{L+1}, \ldots, a_{2L}, \ldots, a_{ML}\}$, and a_j represents the j-th TT&C equipment. There are wn_{ij} visible windows between satellite s_i and TT&C equipment a_j in the planning horizon, and the set is $W_{i,j} = \left\{w_{i,j}^k | k = 0, 1, 2, \ldots, wn_{ij}\right\}$. $w_{i,j}^k$ represents the *k-th* visible window between satellite s_i and TT&C equipment a_j in the planning horizon. For each visible window $w_{i,j}^k \in W_{i,j}$, it can be expressed as $w_{i,j}^k = [ws_{i,j}^k, we_{i,j}^k, ud_{i,j}^k]$, where $ws_{i,j}^k$ and $we_{i,j}^k$ are the start time and end time of the visible window. $ud_{i,j}^k$ is the sign of orbit ascending or descending. $W_i = \left(\cup_{1 \le j \le ML} W_{i,j}\right)$ is the set of all the visible windows of satellite s_i between all the TT&C equipment. The set of all visible windows in the planning horizon is expressed as $W = \cup_{1 \le i \le N, 1 \le j \le ML} W_{i,j}$.

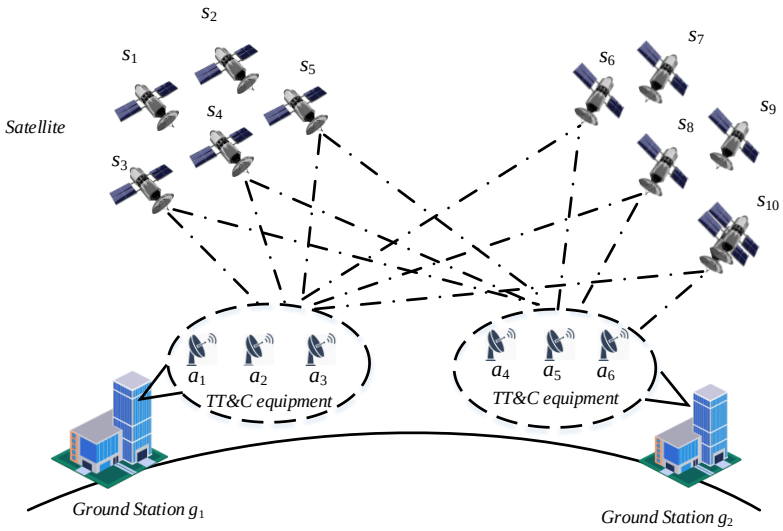


Figure 1. TT&C scenario.

During the planning horizon, all the N satellites in the space network need to be tracked, telemetered and commanded as required. om_i represents the TT&C requirements of the i-th satellite. Therefore, the space network has N TT&C tasks, and their set is denoted as $OM = \{om_i | 1 \le i \le N\}$. For each $om_i \in OM$, it is represented by a four-dimensional tuple $om_i = \{id_i, wr_i, gr_i, \omega_i\}$, where id_i represents the ID of the satellite to be TT&C in task om_i. wr_i represents the requirement of the TT&C window, i.e., the number of TT&C windows required and the request of the orbit ascending or descending for task om_i. gr_i indicates the type of TT&C equipment, that is, the function of the TT&C equipment required by task om_i. ω_i represents the revenue bought by completing task om_i.

As shown in Figure 2, $w_{1,1}^1$, $w_{3,1}^2$, and $w_{2,1}^3$ are three scheduled windows of different tasks performed by the same equipment. A relatively short time interval, i.e., a time fragment, is easily formed between adjacent scheduled time windows of the same TT&C equipment. Specifically, two time intervals with lengths ε_1 and ε_2 are, respectively, formed between the three consecutive time windows $w_{1,1}^1$, $w_{3,1}^2$, and $w_{2,1}^3$ of the TT&C equipment a_1. The length of ε_1 is shorter than the threshold Th, and the length of ε_2 is longer than the threshold Th. Therefore, ε_1 forms a time fragment of equipment a_1. In the planning horizon, if the interval between two adjacent time windows executing tasks in the same TT&C equipment is less than the threshold Th, a time fragment Δt_l is formed, and the set of time fragments is $\Delta T = \{\Delta t_1, \Delta t_2, \Delta t_3, \ldots, \Delta t_l, \ldots\}$.

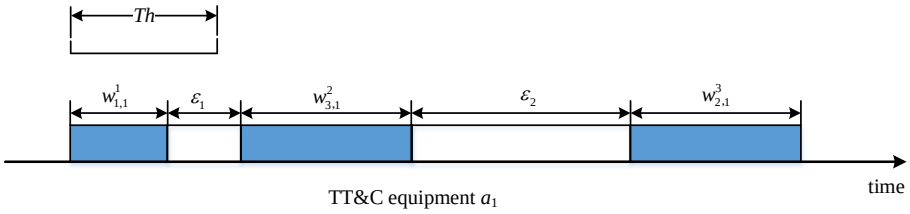


Figure 2. Time fragments.

The essence of the TT&C task-planning problem of the space network is to allocate suitable visible windows of the suitable equipment to each satellite need to be tracked, telemetered and commanded. In this way, the TT&C equipment can track, telemetry, monitor, and control the satellite flight orbit, attitude, and operational status of the onboard subsystems. The core problem studied in this paper is to maximize the total revenue of TT&C tasks while minimizing the total time fragments after the execution of TT&C tasks without exceeding the capacity of network resources.

3. Mathematical Model

During the execution of the TT&C task, the transceiver equipment between the satellite and the ground station can only perform one task at a time. Therefore, there may be conflicts in the simultaneous scheduling of different visible windows. For any two visible windows $w_{i,j}^k, w_{m,n}^r \in W$, they conflict with each other if both of the following conditions are satisfied:

(1) There is time overlap between visible windows $w_{i,j}^k$ and $w_{m,n}^r$, i.e., $ws_{m,n}^r < ws_{i,j}^k < we_{m,n}^r$ or $ws_{m,n}^r < we_{i,j}^k < we_{m,n}^r$.

(2) Windows $w_{i,j}^k$ and $w_{m,n}^r$ occupy the same satellite or TT&C equipment, i.e., $i = m$ or $j = n$.

As shown in Figure 3, the start time of window $w_{2,1}^1$ is before the end time of window $w_{1,1}^1$, and the end time of window $w_{2,1}^1$ is after the start time of window $w_{1,1}^2$. The overlapping part of the two windows together uses the same equipment to cause a conflict.

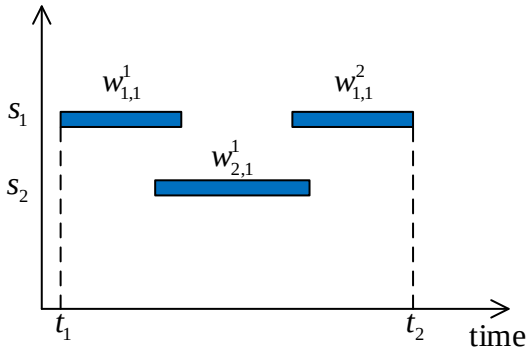


Figure 3. Conflicts among visible windows.

A TT&C task of a satellite requires multiple ascending or descending orbit TT&C windows. The combination of visible windows that meet the requirements constitutes a feasible scheme for the TT&C task. In this paper, y_i^α represents the α-th feasible scheme of task om_i, so the set of feasible schemes for the task om_i is $Y_i = \{y_i^1, y_i^2, \ldots, y_i^\alpha, \ldots\}$, and the set of feasible schemes for all TT&C tasks is $Y = \bigcup_{1 \le i \le N} Y_i$. For example, the window requirement wr_i of task om_i is two ascending orbit windows and two descending orbit windows. The α-th feasible scheme of TT&C task om_i is $y_i^\alpha = \left\{w_i^{\alpha,1}, w_i^{\alpha,2}, w_i^{\alpha,3}, w_i^{\alpha,4}\right\}$, and $w_i^{\alpha,1}, w_i^{\alpha,2}, w_i^{\alpha,3}, w_i^{\alpha,4} \in W$, where $w_i^{\alpha,1}$, $w_i^{\alpha,1}$, $w_i^{\alpha,3}$, and $w_i^{\alpha,4}$ are, respectively, the first ascending orbit window, the first descending orbit window, the second ascending orbit window, and the second descending orbit window in the α-th scheme of the TT&C task.

Since different visible windows between satellites and TT&C equipment may conflict, there may also be conflicts between different feasible schemes. If different feasible schemes contain the same window or if there is a conflict between their visible windows, there is a conflict between the feasible schemes. Therefore, for any two feasible schemes $y_i^\alpha, y_j^\beta \in Y$, a conflict between them must meet one of the following two conditions:

(1) $\exists w_{i,j}^k \in W$ has $w_{i,j}^k \in y_i^\alpha$ and $w_{i,j}^k \in y_j^\beta$.

(2) $\exists w_{i,j}^k \in y_i^\alpha, w_{m,n}^r \in y_j^\beta$, there is a conflict between window $w_{i,j}^k$ and window $w_{m,n}^r$.

Use $O\left(y_i^\alpha\right)$ to represent the set of all feasible schemes that conflict with y_i^α. The Boolean variable $x_i^\alpha = \{0, 1\}$ is used to indicate whether the α-th feasible scheme of task om_i is scheduled. Specifically, $x_i^\alpha = 1$ means that the α-th feasible scheme of task om_i is scheduled. Otherwise, $x_i^\alpha = 0$ means that the α-th feasible scheme of task om_i is not scheduling.

The goal of TT&C task planning is to allocate appropriate time windows for all TT&C tasks within the planning horizon so that the total revenue of TT&C tasks is maximized and the time fragments generated by them are minimized. The following conditions must be met when planning the TT&C tasks:

(1) A TT&C task can only be completed once by a TT&C feasible scheme.

(2) A TT&C scheme and its conflicting scheme cannot be scheduling at the same time.

The low time fragmentation-oriented TT&C task-planning problem is modeled as the following multi-objective optimization problem:

$$\mathrm{P}: \begin{cases} \min \sum_{l=1}^{|\Delta T|} |\Delta t_l| \\ \max \sum_{i=1}^{N} \omega_i \sum_{\alpha=1}^{|Y_i|} x_i^\alpha \end{cases}$$

Subject to the following:

$$\mathrm{C1}: \sum_{\alpha=1}^{|Y_i|} x_i^\alpha \leq 1, \ \forall 1 \leq i \leq N \tag{1}$$

$$\mathrm{C2}: \ x_i^\alpha + \sum_{y_i^\beta \in O(y_i^\alpha)} x_j^\beta \leq 1, \ \forall y_i^\alpha \in Y \tag{2}$$

where C1 is the unique constraint of tasks, which guarantees that each TT&C task can only be completed once at most. C2 is the conflict constraint of the TT&C scheme, i.e., a TT&C scheme and its conflicting schemes cannot be scheduling simultaneously.

4. Problem Transformation

Observing the above optimization problem P, we can see that this problem is a multi-objective combinatorial optimization problem and has been proven to be NP-hard [18]. To solve the TT&C task-planning problem more conveniently and effectively, this paper first transforms the problem into a graph theory problem, and then uses machine learning to solve the problem. Specifically, a task-planning conflict graph $CG(V, E)$ is constructed, and the vertex v_i^α in the graph represents the task feasible scheme y_i^α, i.e., $V = \left\{v_i^\alpha | y_i^\alpha \in Y\right\}$. The edges in CG represent the conflicted relationship between different feasible task feasible schemes. For any $v_i^\alpha, v_n^\beta \in V$, a conflict between them must meet one of the following two conditions:

(1) v_i^α and v_n^β present the two different feasible schemes of the same task, i.e., $i = n$.

(2) There is a conflict between the feasible schemes corresponding to v_i^α and v_n^β. That is to say, there is $y_n^\beta \in O(y_i^\alpha)$ for v_i^α.

It means that there is a conflicted relationship between two vertices, namely, $\left(v_i^\alpha, v_n^\beta\right) \in E$.

The above two conditions correspond to the constraints C1 and C2 in problem P. C1 represents the conflicted relationship between different feasible schemes of the same TT&C task. C2 indicates the conflicted relationship between feasible schemes of different TT&C tasks. Therefore, solving the optimization problem P is equivalent to solving the independent set problem on the graph.

5. Low Time Fragmentation-Oriented TT&C Task-Planning Algorithm

5.1. Graph Learning-Based Framework

Section 4 has transformed the TT&C task-planning problem into the problem of solving the independent set in the task-planning conflict graph. Based on this, this section combines deep learning and graph theory to propose a problem-solving framework, as shown in Figure 4. In this framework, the set *IS* represents the local solution of the independent set problem, which is a subset of set *V*. The Q evaluation function is obtained through reinforcement learning training.

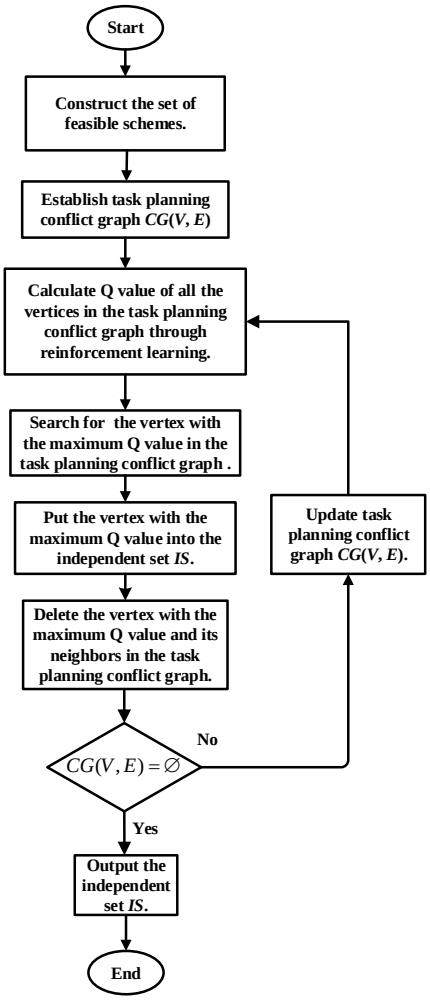


Figure 4. Graph learning-based framework for solving the TT&C task-planning problem.

As shown in Figure 4, we first construct a set of feasible schemes. On this basis, we construct a task-planning conflict graph and calculate the Q values of all vertices in the graph. Then, according to the Q evaluation function trained by deep learning, we select the vertex with the largest Q value. Additionally, we add this selected vertex to the current local solution *IS*. Afterward, we deleted the vertex from the task-planning conflict graph, and the task-planning conflict graph is updated. Lastly, we repeat the above process until there are no remaining vertices in the task-planning conflict graph. The current local solution *IS* is the optimal independent set.

5.2. Feasible Scheme Set Construction

A TT&C task may have many feasible schemes, which increase the complexity of TT&C task planning of space network. To better solve the TT&C task-planning problem of space networks, this paper constructs the feasible scheme set of all tasks in advance. Specifically, we model the feasible planning orders for visible windows of the same task as a directed graph model. Then, this paper transforms the problem of constructing the set of feasible schemes into a path searching problem in directed graph. Finally, we use the traversal search algorithm to find all the feasible schemes of all TT&C tasks.

Let directed graph $G_i(V_i, E_i)$ represent all visible windows of satellite s_i as well as their relationship in a planning horizon. Vertex set $V_i = W_i \cup \{B, E\}$ in the graph represents the vertex set corresponding to all visible windows between satellite s_i and the TT&C equipment within a planning horizon. The start vertex B and the end vertex E are two virtual visible windows of the TT&C task om_i. The directed edge of directed graph G_i represents the sequential scheduling relationship between two visible windows for task om_i. For instance, windows $w_{i,m}^k, w_{i,n}^r \in W$ are two visible windows for task om_i. If there is no overlap between the two windows and the ascending or descending orbit alternate condition is satisfied, the two windows can be scheduled sequentially for task om_i. That is to say, E_i in the graph represents the connection relationship between vertices, which is defined as follows:

(1) $\forall w_{i,j}^k \in V_i$, exists $(B, w_{i,j}^k) \in E_i$, and $(w_{i,j}^k, E) \in E_i$;

(2) $\exists w_{i,m}^k, w_{i,n}^r \in V_i$, if the two windows do not overlap and one is an orbit ascending window and the other is an orbit descending window (without loss of generality, we assume that $ws_{i,n}^r > we_{i,m}^k$), then $(w_{i,m}^k, w_{i,n}^r) \in E_i$.

As shown in Figure 5, vertex B and vertex E represent the virtual start window and virtual end window, respectively. $w_{1,1}^1$, $w_{1,1}^2$, $w_{1,2}^1$, $w_{1,1}^3$, $w_{1,1}^4$, $w_{1,2}^2$, $w_{1,2}^3$, and $w_{1,2}^4$ represent different visible windows, respectively. A path that satisfies the number of windows and the window of ascending or descending requirements represents a feasible scheme within the planning period. Since the aim of TT&C tasks of space networks is to track, telemeter, and command satellites, different tasks are independent of each other in the construction of feasible schemes. That is to say, the directed graphs of each satellite are independent of each other, and the problem of constructing feasible schemes is transformed into the path problem of each independently directed graph. This paper uses the traversal search algorithm on the graph to complete the construction of the feasible scheme set [19–21].

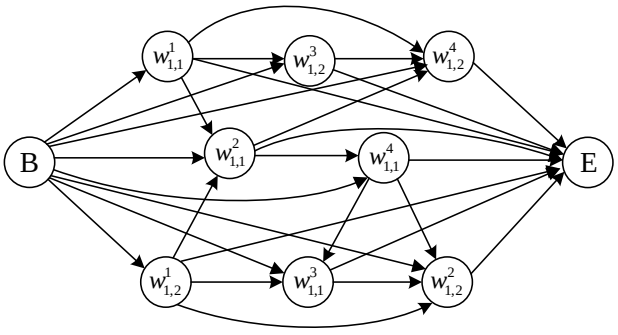


Figure 5. Directed graph of visible windows.

5.3. Vertex Attribute Extraction and Parametric Design of Q Function

Define the vertex attribute $x_v = (d_v, \beta_v, \varpi_v, \omega_v)$ of the task-planning conflict graph, where d_v is the degree of vertex v in the current task-planning conflict graph $CG_{\overline{IS}}$, and β_v is the number of vertices in $CG_{\overline{IS}}$ that correspond to the same task as vertex v in $CG_{\overline{IS}}$, namely,

$$\beta_v = \left| \{ v_i^\alpha | v_i^\alpha \in V_{\overline{IS}} \} \right| \tag{3}$$

where the operator $|\cdot|$ is the number of elements in the set. ω_v represents the newly added time fragments after the feasible scheme represented by vertex v is added to the current local solution. ω_v is the revenue obtained after the task feasible scheme corresponding to vertex v is completed.

In order to simplify the symbolic expression, v is used in the following text to express vertex v_i^α in the task-planning conflict graph. Graph structure embedding is carried out recursively. In each step of the recursive process, information is embedded for each vertex of the graph, so that it contains a larger range of graph structure and vertex attribute information. By embedding different degrees of information on each vertex, the computation can be effectively reduced and the algorithm efficiency can be improved. In particular, in the recursive process, a q-dimensional vector $\mu_v^{(k)}$ is used to represent the embedding vector of the vertex v obtained after the k-th iteration. The initial value of the embedding vector of vertex v is set to 0.

$$\mu_v^{(k+1)} \leftarrow relu(\theta_1 x_v + \theta_2 \sum_{u \in N_{\overline{IS}}(v)} \mu_u^{(k)}) \tag{4}$$

It can be seen from Equation (4) that the updating process of graph structure embedding is based on the adjacency relationship of vertices in the graph, where $N_{\overline{IS}}(v)$ is the vertices adjacent to the vertex v in the $CG_{\overline{IS}}$ graph. θ_1 and θ_2 are the model parameters, and *relu*() is the rectified linear unit. Based on Equation (4), one can see that the embedding update process is carried out based on the graph topology. A new round of embedding sweeping across the vertexes will start only after the embedding update for all nodes from the previous round has finished. The above process goes through a total of K iterations, and each vertex obtains an embedding vector $\mu_v^{(K)}$.

After the K-step graph embedding information is completed, the embedding vector is defined as the evaluation function $\mathrm{Q}(IS, v; \Theta)$. $\sum_{u \in IS} \mu_u$ and μ_v are used to represent the graph CG and vertex v, respectively, and $\mathrm{Q}(IS, v; \Theta)$ is defined as follows:

$$\mathrm{Q}(IS, v; \Theta) = \theta_3^{\mathrm{T}} relu[\theta_4 \sum_{u \in IS} \mu_u, \theta_5 \mu_v] \tag{5}$$

where θ_3^{T}, θ_4, and θ_5 are the parameters to be trained for the model, and [.,.] is the concatenation operator. The parameters μ_u and μ_v are calculated by Equation (4). They are affected by the parameters θ_1 and θ_2 in the embedding process of the graph structure. So the evaluation function $\mathrm{Q}(IS, v; \Theta)$ is determined by five parameters $\{\theta_i\}_{i=1}^5$.

Algorithm 1 shows the specific calculation steps of the evaluation function $\mathrm{Q}(IS, v; \Theta)$, where $\sum_{u \in IS} \mu_u^k$ needs to be recalculated according to the task-planning the subgraph of task planning conflict graph at that time in each iteration. The parameters $\{\theta_i\}_{i=1}^5$ are obtained through deep Q-learning training.

Algorithm 1 Calculation of evaluation funtion $\mathrm{Q}(IS, v; \Theta)$

1: Initialize x_v
2: **for** k = 1 to K do
3: **for** $v \in V$ do
4: $\mu_v^{(k+1)} \leftarrow relu(\theta_1 x_v + \theta_2 \sum_{u \in N_{\overline{IS}}(v)} \mu_u^{(k)})$
5: **end for**
6: **end for**
7: Return $\mathrm{Q}(IS, v; \Theta) = \theta_3^{\mathrm{T}} relu[\theta_4 \sum_{u \in IS} \mu_u, \theta_5 \mu_v]$

5.4. Evaluation Function Based on DDQN Network Training

As a typical representative of the value-based algorithm in the reinforcement learning algorithm, the DQN algorithm combines deep learning and reinforcement learning to predict the Q value. Q value can be understood as the value of state action, i.e., the expected revenues of the agent through acting in a certain state. Although the DQN algorithm

has achieved good results in deep reinforcement learning, it sometimes overestimates the behavior value too optimistically. This will cause the algorithm to unanimously regard the suboptimal behavior value as the optimal behavior value, eventually failing to converge to the best value function. However, the calculation method of the target value of DDQN and DQN is different. In DQN, the largest Q value in the target network is directly selected as the target value. Additionally, DDQN calculates the Q value according to the parameters of the current Q network. Therefore, the Q value in DDQN is less than or equal to the Q value in DQN. This reduces the overestimation to a certain extent, making the Q value closer to the real value.

Therefore, in this section, solving the independent set in the task-planning conflict graph is modeled as a DDQN learning process. Additionally, the Q evaluation function has been trained. This paper defines the state, action, and reward under the reinforcement learning framework. The specific content is as follows:

1. State: The state in reinforcement learning is the local solution IS. Because the impact of local solution on the subsequent independent set construction process is equal to the task-planning conflict graph in the iteration process, the q-dimension embedded vector $\sum_{u \in IS} \mu_u$ of the task-planning conflict subgraph is used to represent the current state.
2. Action: The action v is a vertex that does not belong to the current local solution IS. Additionally, the action v is expressed as a q-dimensional embedding vector μ_v.
3. Reward: This paper aims to minimize time fragments and maximize task revenue. Therefore, the designed reward is the revenue of vertex v minus the total time fragment's increment after vertex v is added to the current local solution IS, namely:

$$r(IS, v) = \lambda \omega_v - \Delta T(IS, v) \tag{6}$$

where $\Delta T(IS, v)$ is the total time fragment generated by adding the feasible scheme represented by vertex v to the set of feasible schemes corresponding to the current solution IS, and $\lambda \in (0,1)$ is a constant.

The evaluation function $\mathrm{Q}(IS, v; \Theta)$ gives the value of action v in a given state IS. DDQN is used to optimize parameter Θ of the evaluation $\mathrm{Q}(IS, v; \Theta)$ function in the reinforcement learning process.

In this paper, two DDQN neural networks are constructed to train model parameters. i.e., target network and evaluation network. The evaluation network is used to select an action, and the target network is used to evaluate the value of the action. The two networks update parameters alternately with each other. Specifically, target network has parameter Θ_t and evaluation network has parameter Θ_e are established. Additionally, the parameter Θ_e is updated by minimizing the square loss through the gradient descent method:

$$L(\Theta) = E\left[\left(h - \mathrm{Q}(IS, v; \Theta_e)\right)^2\right] \tag{7}$$

During the training process, the neural network copies parameter Θ_t to parameter Θ_e every several step. In DDQN, the target value is defined as

$$h = \begin{cases} r(IS_\tau, v_\tau) + \gamma \mathrm{Q}(IS_\tau, \underset{v'}{\mathrm{argmax}} \mathrm{Q}(IS_{\tau+1}, v', \Theta_e); \Theta_t) & IS \neq \widehat{IS} \\ r(IS_\tau, v_\tau) & IS = \widehat{IS} \end{cases} \tag{8}$$

where γ is the Q-learning decay coefficient.

When constructing an independent set, the reward obtained for each additional vertex is not only from the successful planning of the task represented by this vertex, but also from subsequent task rewards in non-conflicting task planning. For this delayed reward scenario, the Q function updated at each step is too short-sighted to reflect the long-term revenue situation well. Therefore, this paper uses the n-step Q-learning method to update the parameters. Unlike single-step learning, n-step Q-learning needs to wait n steps for

parameter update. Only in this way can the Q-function provide a more precise evaluation of future rewards. The parameter update method is the same as the single-step learning (Equation (7)). Therefore, there is a difference in the calculation of the target value:

$$h = \begin{cases} \sum_{i=1}^{n-1} r(IS_{\tau+i}, v_{\tau+i}) + \gamma Q(IS_{\tau+i}, \underset{v'}{\text{argmax}} Q(IS_{\tau+n}, v', \Theta_e); \Theta_t) & IS_{\tau+n-1} \neq \widehat{IS} \\ \sum_{i=1}^{n-1} r(IS_{\tau+i}, v_{\tau+i}) & IS_{\tau+n-1} = \widehat{IS} \end{cases} \quad (9)$$

Moreover, there is a strong correlation between the continuous actions of the independent set construction process. In order to reduce the impact of the correlation between continuous samples on neural network training, this paper uses experience replay technology to speed up the convergence speed of neural network and improve the robustness of training results. Specifically, samples are continuously added to the data set Π at each step in every round. For example, in step $\tau + n$, the tuple $(IS_{\tau+n}, v_{\tau+n}, R_{\tau+n,\tau}, IS_\tau)$ is added to Π, where $R_{\tau+n,\tau} = \sum_{i=1}^{n-1} r(IS_{\tau+i}, v_{\tau+i})$.

In summary, we find that there is a reward delay problem during independent set construction and that there is a strong correlation between consecutive samples. Based on this, we first address the reward-delay situation during independent set construction using n-step Q-learning. Then, we use the experience replay technique to reduce the correlation between consecutive samples. At last, this paper designs an independent set algorithm based on double deep Q-learning to complete the training of parameter Θ, as shown in Algorithm 2. $CG(V_c, E_c)$ is a task-planning conflict graph constructed from a set of tasks drawn from the sample space.

Algorithm 2: Evaluation function training based on DDQN learning

Initialize experience replay storage space Π.
for round e = 1 to L do
  Extract a task set and construct the corresponding task-planning conflict graph $CG(V_c, E_c)$
  Initialize τ = 1, IS_1 = Ø, and random parameters Θ_t, Θ_e.
  while $V_{\overline{IS_\tau}} \neq \varnothing$:
    $v_\tau = \begin{cases} \text{random node } v \in V_{\overline{IS_{\tau'}}} & \text{w.p.} \varepsilon \\ \text{argmax}_{v \in V_{\overline{IS_\tau}}} Q(IS_\tau, v; \Theta_e) & \text{w.p.} 1-\varepsilon \end{cases}$.
    Add vertex v_t into the local solution: $IS_{\tau+1} = IS_\tau \cup \{v_\tau\}$
    Delete vertex v_τ and its neighbor vertices from $CG_{\overline{IS}}$.
    if $\tau \geq$ n then
      Add tuple $(IS_{\tau+n}, v_{\tau+n}, R_{\tau+n,\tau}, IS_\tau)$ to Π.
      Sample random batch from B~Π.
      Use the samples in B to train and update the value of Θ_e based on the gradient descent method.
      Replace target parameters $\Theta_t \leftarrow \Theta_e$ every c steps
    end if
    $\tau = \tau + 1$.
  end while
end for
return Θ_t

6. Simulation

In this paper, STK11.3 software is used to build a scenario of space networks. Tensorflow1.15 is used to build a neural network, and Python3.7 is used to build the algorithm main program. The space network scenario configuration is as follows:

(1) A total of 150 low-orbit satellites are located in 150 different orbits. The orbital altitudes are 500 km, 650 km, and 800 km. Inclinations are evenly distributed within the range of 96 to 100 degrees.

(2) There are two ground stations (Xi'an Station and Kashi Station in China), each of which has three sets of TT&C equipment.

The task-planning horizon is from 2022-8-27 04:00:00 to 2022-8-28 04:00:00. During the experiment, each TT&C task has two to six different TT&C window requests of the orbit ascending or descending. A total of 150 groups of tasks are randomly generated, of which 120 are used as training sets and 30 as test sets. The time fragment length threshold Th is set to 5 min during the experiment.

Figure 6 shows the convergence of training the function Q network at different learning rates. It can be seen that the larger the learning rate, the faster the Q-function converges. When the learning rate is 0.05, the loss function converges the fastest, but the result quickly falls into the local optimum. When the learning efficiency is 0.0005, the convergence speed of the loss function is too slow, resulting in too long a training time to achieve a good training effect. Therefore, this paper uses a learning rate of 0.005 to train the Q evaluation function. Because this can not only prevent the loss function from converging too slowly but also prevent the loss function from falling into a local optimum to a certain extent. Finally, the best training result is achieved.

In order to verify the performance of the algorithm proposed in this paper, this paper compares the LTFTT&C algorithm with the randomized adaptive search procedure (GRASP) [22] and genetic algorithm (GA) [23]. It mainly conduct performance comparisons in terms of the total time fragments in different TT&C task scenarios, average response time of emergency tasks, and total task revenue.

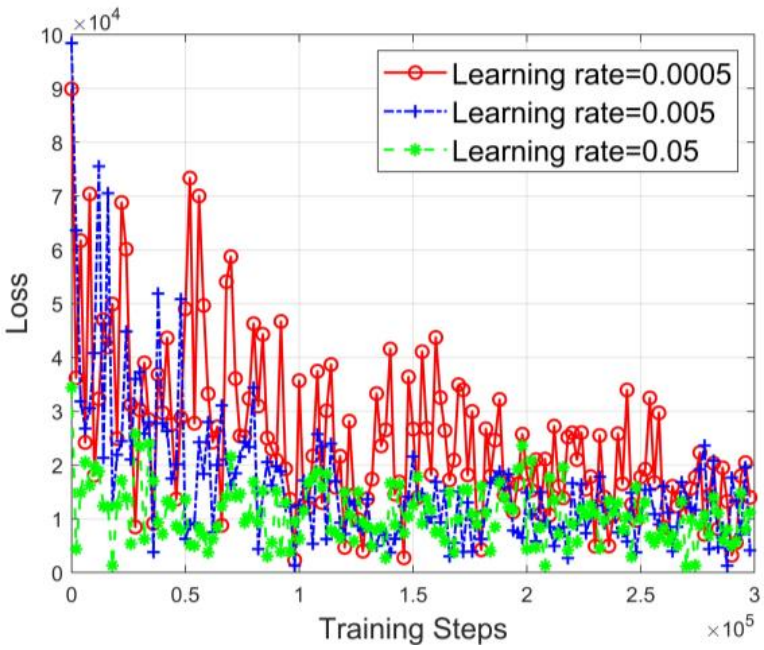


Figure 6. The change graph of loss function with the number of training steps.

Figure 7 compares the total time fragments of the TT&C equipment of the three TT&C task-planning algorithms under different task numbers. It can be seen that, compared with the other two algorithms, the LTFTT&C algorithm significantly reduced the total amount of time fragments of equipment under different task numbers. Compared with the GRASP algorithm, LTFTT&C reduced the total time fragments of TT&C equipment by 48%. Compared with the genetic algorithm, LTFTT&C reduced the total time fragments of TT&C equipment by 32%. To further verify the LTFTT&C algorithm to reduce the gain of time fragmentation on emergency task planning, the emergency task test response time (i.e., the difference between the emergency task completion time and arrival time) was inserted based on the three algorithms for common TT&C tasks. As shown in Figure 8, with the increase in the number of emergency tasks, the average response time of the emergency

tasks of the three algorithms increases to varying degrees, and the LTFTT&C algorithm is obviously better than the other two algorithms. LTFTT&C is 43% faster than the GRASP algorithm and 36% faster than the genetic algorithm in the average response time of the emergency tasks. This is mainly because the LTFTT&C algorithm effectively reduces the amount of time fragments, making it easier for emergency tasks to insert into suitable visible windows.

Figure 9 shows the performance of the three algorithms regarding the total task revenue. It can be seen from the figure that when the number of tasks is small, the LTFTT&C algorithm is significantly better than the other two comparison algorithms in terms of task revenue. It is 21% and 15% higher than the GRASP and genetic algorithms. When the number of tasks is large, the LTFTT&C algorithm improves the total task revenue by 28% compared with the GRASP algorithm. It decreases the total task revenue by 7% compared with the genetic algorithm. Combining Figures 7 and 9, compared with GRASP, the LTFTT&C algorithm adopts the deep reinforcement learning method, which not only improves the revenue but also reduces the time fragment. Compared with the genetic algorithm, the LTFTT&C algorithm sacrifices a small amount of revenue in exchange for a significant decrease in time fragmentation, thus effectively reducing the response time of emergency tasks.

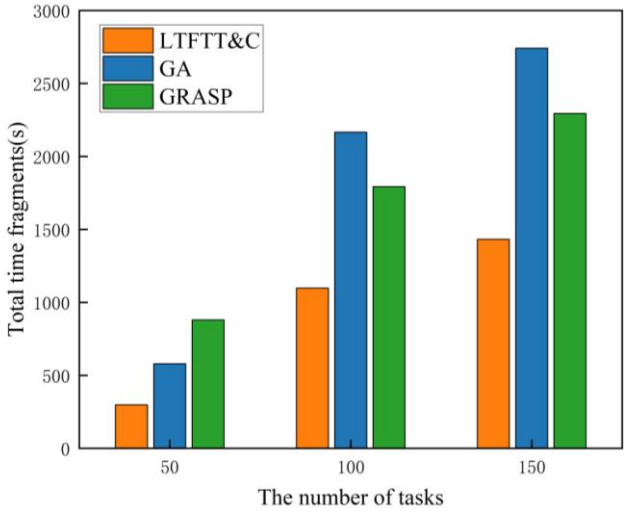


Figure 7. Total time fragments vs. number of tasks.

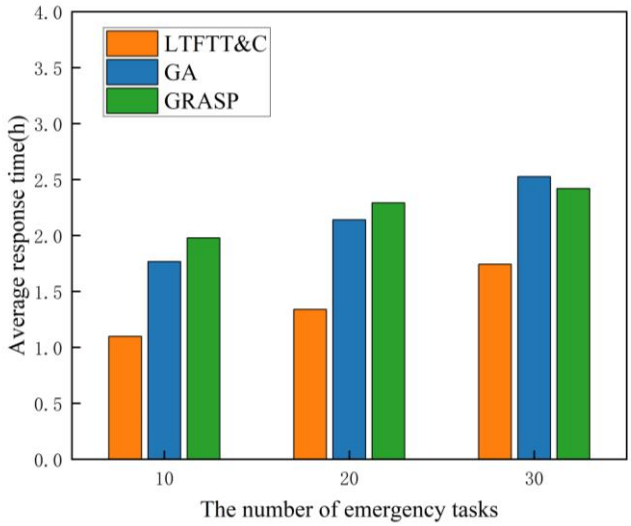


Figure 8. Average response time of emergency tasks.

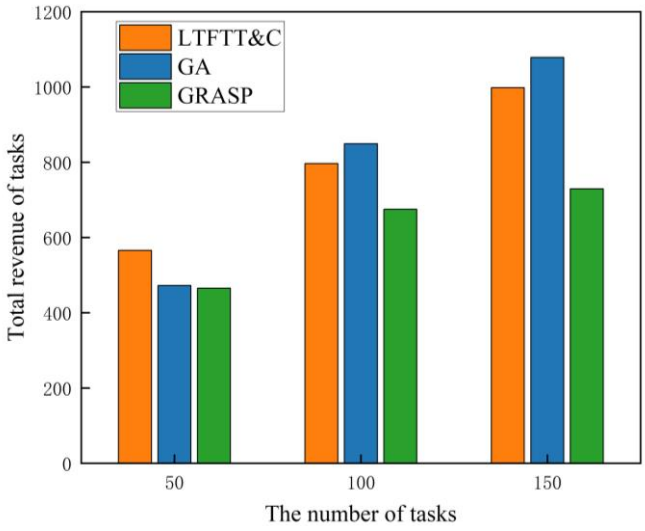


Figure 9. Total revenue of tasks vs. number of tasks.

7. Conclusions

In this paper, we proposed a low time fragmentation-oriented TT&C task-planning method to reduce the total time fragments of TT&C equipment and improve the response time of emergency tasks. Firstly, this paper established a mathematical model with the optimization goal of minimizing the total time fragments and maximizing the total revenue of TT&C tasks. Then, we transformed the problem of TT&C task planning into an independent set problem in the graph, and introduced DDQN in deep reinforcement learning as an evaluation strategy. Afterward, we designed a low time fragmentation-oriented TT&C task-planning method to improve the service capability of TT&C resources. The simulation results of multiple experimental scenarios showed that the LTFTT&C algorithm is significantly better than GRASP and GA algorithms in terms of time fragmentation. Therefore, using the LTFTT&C algorithm to solve the TT&C task-planning problem is reasonable and effective.

Author Contributions: Methodology, H.X. and R.L.; Validation, H.X.; Writing—original draft, H.X.; Writing—review & editing, R.L.; Supervision, R.L. All authors contributed equally to this work. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the Natural Science Foundation of Shaanxi Province (2023-JC-YB-556), in part by the Young Talent fund of University Association for Science and Technology in Shaanxi China (20200112), in part by the Key Research and Development Program in Shaanxi Province of China (2021GY-006), and in part by the Postdoctoral Foundation in Shaanxi Province of China (2018BSHEDZZ47).

Institutional Review Board Statement: The study did not require ethical approval.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Conflicts of Interest: The author declares no conflict of interest.

References

1. Zhan, Y.; Wan, P.; Jiang, C.; Pan, X.; Chen, X.; Guo, S. Challenges and solutions for the satellite tracking, telemetry, and command system. *IEEE Wirel. Commun.* **2020**, *27*, 12–18. [CrossRef]
2. Chen, S. Analysis on low-orbit satellite communication and my country's development suggestions. *Telecommun. Sci.* **2020**, *36*, 1–13.
3. Dong, Y.; Li, Y.; Chen, J.; Ou, T.; Li, Y.; Xi'an Satellite Control Center. Demand analysis of command control system of the space TT&C network. In Proceedings of the 2nd International Conference on Information Technology and Computer Application (ITCA), Beijing, China, 25 July 2019; pp. 236–239.

4. Zhou, D.; Sheng, M.; Li, J.; Han, Z. Aerospace Integrated Networks Innovation for Empowering 6G: A Survey and Future Challenges. *IEEE Commun. Surv. Tutor.* **2023**. [CrossRef]
5. Zhai, X.; Niu, X.; Tang, H.; Wu, L.; Shen, Y. Robust satellite scheduling approach for dynamic emergency tasks. *Math. Probl. Eng.* **2015**, *2015*, 1–20. [CrossRef]
6. Dai, C.Q.; Li, C.; Fu, S.; Zhao, J.; Chen, Q. Dynamic scheduling for emergency tasks in space data relay network. *IEEE Trans. Veh. Technol.* **2020**, *70*, 795–807. [CrossRef]
7. Whitley, D.; Sutton, A.; Howe, A.; Barbulescu, L. Resource scheduling with permutation based representations: Three applications. *Evol. Comput. Pract.* **2008**, *88*, 219–243.
8. Marinelli, F.; Nocella, S.; Rossi, F.; Smriglio, S. A Lagrangian heuristic for satellite range scheduling with resource constraints. *Comput. Oper. Res.* **2011**, *38*, 1572–1583. [CrossRef]
9. Sarkheyli, A.; Bagheri, A.; Ghorbani-Vaghei, B.; Askari-Moghadam, R. Using an effective tabu search in interactive resources scheduling problem for LEO satellites tasks. *Aerosp. Sci. Technol.* **2013**, *29*, 287–295. [CrossRef]
10. Chen, M.; Wen, J.; Song, Y.J.; Xing, L.N.; Chen, Y.W. A population perturbation and elimination strategy based genetic algorithm for multi-satellite TT&C scheduling problem. *Swarm Evol. Comput.* **2021**, *65*, 100912.
11. Ren, B.; Liu, J.; Li, Z.; Wu, J.; Yao, Z. Satellite Requirement Preference Driven TT&C Resources Scheduling Algorithm for Time Sensitive Tasks. In Proceedings of the 3rd International Conference on Electronic Information and Communication Technology (ICEICT), Shenzhen, China, 13–15 November 2020; pp. 15–19.
12. Gu, X.; Bai, J.; Zhang, C.; Gao, H. Study on TT&C resources scheduling technique based on inter-satellite link. *Acta Astronaut.* **2014**, *104*, 26–32.
13. Zhang, Z.; Hu, F.; Zhang, N. Ant colony algorithm for satellite control resource scheduling problem. *Appl. Intell.* **2018**, *48*, 3295–3305. [CrossRef]
14. Zhang, N.; Feng, Z.; Ke, L. Guidance-solution based ant colony optimization for satellite control resource scheduling problem. *Appl. Intell.* **2011**, *35*, 436–444. [CrossRef]
15. Zhang, Z.; Zhang, N.; Feng, Z. Multi-satellite control resource scheduling based on ant colony optimization. *Expert Syst. Appl.* **2014**, *41*, 2816–2823. [CrossRef]
16. Wang, Y.; Zhou, D.; Song, N.; Sheng, M.; Li, J.; Liu, J. Concurrent Reconfiguration of Resource-Oriented Emergency TT&C Task Scheduling for Space Information Networks. *J. Commun. Inf. Netw.* **2021**, *6*, 142–152.
17. Xue, N.; Ding, D.; Fan, Y.; Wang, Z. Research on Joint Scheduling Method of Heterogeneous TT&C Network Resources Based on Improved DQN Algorithm. In Proceedings of the 2nd International Conference on Information Technology, Big Data and Artificial Intelligence (ICIBA), Chongqing, China, 17–19 December 2021; Volume 2, pp. 1009–1014.
18. Barbulescu, L.; Watson, J.P.; Whitley, L.D.; Howe, A.E. Scheduling space–ground communications for the air force satellite control network. *J. Sched.* **2004**, *7*, 7–34. [CrossRef]
19. Christofides, N. The optimum traversal of a graph. *Omega* **1973**, *1*, 719–732. [CrossRef]
20. Cheung, T.Y. Graph traversal techniques and the maximum flow problem in distributed computation. *IEEE Trans. Softw. Eng.* **1983**, *SE-9*, 504–512. [CrossRef]
21. Buchsbaum, A.L.; Goldwasser, M.H.; Venkatasubramanian, S.; Westbrook, J.R. On external memory graph traversal. In Proceedings of the ACM-SIAM Symposium on Discrete Algoritma, San Francisco, CA, USA, 9–11 January 2000; pp. 859–860.
22. Rojanasoonthon, S.; Bard, J. A GRASP for parallel machine scheduling with time windows. *INFORMS J. Comput.* **2005**, *17*, 32–51. [CrossRef]
23. Wu, B.; Li, Y.-X.; Huang, Y.-X. Optimal scheduling of TT&C network resources based on genetic algorithm. *J. Astronaut.* **2006**, *27*, 1132–1136.

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.