

Article

A Satellite Task Planning Algorithm Based on a Symmetric Recurrent Neural Network

Sikai Liu * and Jun Yang

College of Intelligence Science and Technology, National University of Defense Technology, Changsha 410073, China; jyang@nudt.edu.cn

* Correspondence: liusikai08@nudt.edu.cn; Tel.: +86-185-7310-4146

Received: 16 October 2019; Accepted: 3 November 2019; Published: 6 November 2019



Abstract: The intelligent satellite, iSAT, is a concept based on software-defined satellites. Earth observation is one of the important applications of intelligent satellites. With the increasing demand for rapid satellite response and observation tasks, intelligent satellite in-orbit task planning has become an inevitable trend. In this paper, a mixed integer programming model for observation tasks is established, and a heuristic search algorithm based on a symmetric recurrent neural network is proposed. The configurable probability of the observation task is obtained by constructing a structural symmetric recurrent neural network, and finally, the optimal task planning scheme is obtained. The experimental results are compared with several typical heuristic search algorithms, which have certain advantages, and the validity of the paper is verified. Finally, future application prospects of the method are discussed.

Keywords: in-orbit observation; task planning; recurrent neural network; heuristic algorithm

1. Introduction

1.1. Proposal of the Intelligent Satellite

The various existing types of application satellites already have a strong performance. On the one hand, the communication ability, remote sensing, navigation, and other payloads of these satellites have been greatly improved, and they now basically meet the needs of military activities and civil fields. On the other hand, the life of satellites in orbit has been prolonged. However, once these satellites are launched into orbit, the performance parameters of their loads cannot be changed throughout the life of the device, which restricts the effective application of satellites to some extent. At the same time, compared with the open Internet ecological environment, the aerospace industry is relatively closed, and its research and development is basically based on hardware and software. A more common practice is to customize a satellite for a specific function or task. The process has a long development cycle and high research and development costs. Different satellite models are not compatible with each other in terms of hardware, and the components cannot be interchanged. Meanwhile, they are incompatible with each other in terms of software, and software developed for one model cannot be directly run on another satellite model. Developers cannot develop and deploy software for different satellite models. This hardware-based, software-assisted R&D process has become one of the important bottlenecks restricting the development of space technology [1,2]. With the rapid advancement of satellite hardware and software technology, as well as the rapid promotion of software definition ideas, satellites are gradually developing toward miniaturization, a low cost, and a rapid design and deployment. Each country has made software-defined satellites and loads an important development direction. A lot of manpower and resources have been invested in this.

A software defined radio (SDR) satellite is a space-based application extension of the SDR technology [3]. It can reconfigure some functions of a satellite through a software update. Since the concept of the software radio was proposed in the 1990s, the US Department of Defense and the National Aeronautics and Space Administration (NASA) have conducted long-term research, according to their respective needs, and formed corresponding standard specifications, mainly including the Software Communication Architecture (SCA) and Space Telecommunication Radio System (STRS) [4]. Both SCA and STRS standards use an open architecture that supports the decoupling of applications from the platform and supports the connection and communication of application components. STRS has become the first space-based software-defined radio technology standard for space applications that uses a lightweight software architecture design and has developed space-based applications, such as the NASA United General Dynamics Corporation and Harris Corporation's "Space Communication and Navigation" (SCaN) test platform project, and three STRS-based software defined radio loads have subsequently been developed [5]. The European Space Agency proposed the development of the world's first "quantum" satellite, with software-defined payloads, to achieve an in-orbit reconfiguration of coverage areas, frequency bands, bandwidth, and power [6]. In 2019, Loma announced that it will launch a micro-nano software-defined satellite, called "SmartSat" [7]. The satellite uses technologies and new architectures, such as virtual machines and multi-core processing, to significantly increase the speed of response, flexibility and in-orbit processing capabilities, as well as support satellites to change their presence in orbit tasks.

Yang Xiaoniu also proposed the concept of the "Software Satellitem", which means that the payload of such a satellite is a satellite system that uses the software radio, as a common platform, to define its functions through software [8]. In 2017, the Institute of the Chinese Academy of Sciences and the Association of Software-Defined Satellite Technologies, initiated by more than 20 units, launched the "Tianzhi No.1" satellite, based on space-based cloud computing, with an open system architecture and powerful real-time information processing and massive data storage capabilities [9].

These new conceptual satellites have not solved the problems associated with a long design and manufacturing cycle, high cost, weak universality, single application mode, and function, and they cannot meet the development requirements of a space-based network system. Yang Jun proposed the concept of an intelligent satellite, "iSAT", based on software-defined satellite research [10,11]. The goal of iSAT study is to change the traditional satellite service model, from functional to service-oriented customization, standardizing the design and manufacture of satellites through, for instance, designing and manufacturing computers and decoupling user and developer processes. The iSAT satellite can run different application software, according to different task requirements, provide corresponding in-orbit services, share satellite resources in real time, and realize one-satellite multi-energy, multi-satellite coordination, and intelligent services.

The core idea of the iSAT concept is a general satellite architecture. Based on this, research and mass production will be carried out, and finally, there will be multiple intelligent satellites with complete functions in orbit. Each iSAT satellite is a node of the space-based information network. In the process of intelligent satellite application, the underlying physical resources of the satellite are abstracted, and the resource pool is built in the satellite network to separate the infrastructure from the service, so that the resources can be flexibly and efficiently matched, according to the user's business requirements. It is foreseeable that the emergence of iSAT will change the satellite's resource application model, from the traditional task-specific design satellite function load to the multi-satellite in-orbit resource service, for specific tasks.

1.2. Task Planning for Satellite Earth Observation

Intelligent satellites are oriented toward future space-based information network applications and have more comprehensive space observation requirements than traditional satellites. Therefore, Earth observation capability has become one of the core capabilities that must be considered in relation to the development of intelligent satellites. The observation process of the traditional satellite is roughly

as follows: first, the imaging request is given; the imaging satellite ground task observation system is planned, according to the obtained imaging task attribute, satellite information and corresponding constraint conditions; then, the remote sensor control command is generated, according to the result. After the confirmation is found to be correct, instructions are sent to the satellite, and the instructions are executed by the satellite. After the image is transmitted, processed, etc., the image is handed over to the user to complete all the processes. The command is executed by the satellite, the image is sent to the user, after downloading and processing, and all of the processes are completed. Moreover, this “ground planning, on-board execution” approach manages in-orbit satellites and relies too much on the manual development of detailed task execution plan sequences. The amount of command data required for satellites to be greatly increased and the pressure on the ground monitoring and control stations to control the orbiting satellites continue to grow, and they will not be able to adapt to the increasingly precise and complex observation task requirements and high dynamics of intelligent satellite applications. The autonomous satellite observation process is shown in Figure 1.

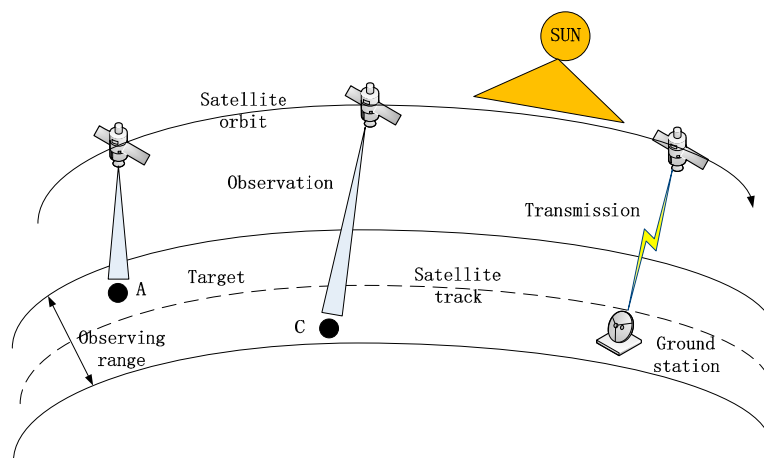


Figure 1. Autonomous satellite observation process.

Due to the manpower consuming and error-prone way of formulating detailed instructions, as well as the development of artificial intelligence technology and space science and technology, autonomous task planning has become one of the satellite intelligent control technologies that many countries have attached importance to in recent years. Autonomous task planning refers to the process in which a satellite, after receiving high-level instructions sent from the ground to describe the task requirements, generates the necessary action sequence with a time stamp in-orbit, according to the current state of the satellite, the action set involved in the satellite task and the flight rules through a certain programming solution algorithm. Satellites with autonomous task planning capabilities can perform various routine operations without interference and have the ability to adapt to fault conditions or unknown environments, self-tolerance, and self-recovery. With the introduction and development of iSAT technology, we will face the rapid growth of users and tasks, continuous and irregular tasks, the dynamic access of ground stations, and a complex computing environment in the satellite. How to maximize the use of satellites and measurement and control resources to meet the needs of dynamically changing task sets is an important reason for studying satellite task planning.

At present, research on the autonomous observation task planning on satellites has achieved certain results, mainly using heuristic search algorithms. However, the application of machine learning methods to satellite autonomous task planning is still in the exploratory stage. Combining the neural network method and heuristic search algorithm, this paper proposes a task planning algorithm, which is superior to the existing heuristic search algorithm in terms of the overall profit. The rest of this paper is organized as follows. Section 2 reviews the existing satellite task planning algorithms and the application of neural networks in satellite task planning. Section 3 describes the problem of satellite in-orbit observation task planning and sets up the research hypotheses and constraints. In Section 4,

the task planning problem is expressed as a mixed integer programming model, and a heuristic search algorithm based on a symmetric recurrent neural network is proposed in Section 5. Section 6 introduces the experimental setting, compares the proposed algorithm with several advanced heuristic algorithms, and concludes that the proposed algorithm outperforms several other algorithms. Finally, Section 7 presents a discussion and analysis, and the application of this method to iSAT satellites is explained.

2. Related Research

2.1. Overview of Satellite Task Planning Algorithms

The task planning problem of intelligent satellites is the same as that of traditional satellites. It is essentially an NP-hard constraint satisfaction problem [12]. In other words, with the goal of satisfying on-board resource constraints and task constraints, the way to arrange the execution sequence and specific execution time of a group of tasks, so as to make one or more objective functions, such as the largest observation area, the largest number of images, the highest energy distribution efficiency, and the largest task income reach, the optimum is determined. The on-board resource constraints include electrical energy, mobility, communication bandwidth, storage space, side viewing capabilities, and so on. The task constraints include the target area (point and face) size and position, sun height angle, target line of the sight angle, and so on. Many researchers have now studied autonomous satellite task planning and tend to use intelligent optimization algorithms and heuristic algorithms to solve them.

The intelligent optimization algorithm does not need to construct an accurate mathematical search direction and does not need to perform complicated one-dimensional search. Instead, it obtains the optimal solution of the problem through a large number of simple information dissemination and evolution methods, including the genetic algorithm, simulated annealing algorithm, tabu search algorithms, etc. [13]. Mouw et al. [14] studied the task planning problem of optical imaging satellites using the tabu search algorithm and divided the process of the solution into stages. Xhafa et al. [15] also designed an adaptive mutation probability genetic algorithm, when studying the task planning problem of Earth observation satellites. Sarkheyli et al. [16], in a study of low-orbit satellite task planning, took task priority, resource constraints and user satisfaction into consideration and designed a new tabu search algorithm to solve them. Ruan et al. [17] adopted the greedy algorithm for task planning and solving. Globus et al. [18] used the simulated annealing algorithm to solve the satellite point target observation task planning problem. He et al. [19] designed a combination algorithm to solve the problem of task planning for an earth observation satellite in order to avoid the instability problem associated with providing a solution using a single traditional algorithm. In order to improve the efficiency of remote sensing satellite scheduling, Gao et al. [20] designed the ant colony algorithm for remote sensing satellite ground integrated task planning. Jiang and Pang [21] designed an adaptive ant colony algorithm by designing a task merging mechanism to solve the problem of huge user requests. Zheng et al. [22] combined the re-planning technique with the multi-objective hybrid dynamic mutation genetic algorithm for dynamic re-planning scenarios of different urgency levels and designed the cyclic re-planning algorithm and the near-real-time re-planning algorithm. Niu [23] used genetic algorithms to solve dynamic task planning problems.

The heuristic algorithm provides a feasible solution to the combinatorial optimization problem, with an acceptable cost (calculation time, space occupation, etc.). The degree of deviation of the feasible solution from the optimal solution is generally not expected. With the deepening of research, heuristic algorithms can also be simple or complex, single or mixed. It has gradually become an effective method for solving combinatorial optimization problems.

Marinelli et al. [24] regards satellite task planning as an integer programming problem. Using a non-standard blackball heuristic algorithm, a heuristic algorithm based on Lagrangian relaxation is proposed to solve the problem and verify the efficiency of the algorithm. Bina et al. [25] proposed a local search algorithm based on heuristic rules. Tangpattanakul et al. [26] designed a heuristic algorithm to solve the multi-satellite multi-orbit task planning problem, with the goal of satisfying

user requests and various observation constraints. Chen et al. [27] designed a heuristic neighborhood search algorithm to solve the problem of satellite and ground receiving station scheduling problems. When Xue [28] studied the task planning of the Earth observation satellite, he used a two-stage planning algorithm based on relaxation map planning, which combines a heuristic search with plan evaluation technology, and obtained a time-stamped sequential planning solution. Si [29] designed an improved heuristic search algorithm, combining the conflict-oriented algorithm and the arc-compatible algorithm to improve the efficiency of the solution. Liu et al. [30] proposed a rolling plan heuristic algorithm, which uses successive multiple local plans, instead of one global plan. Wang et al. [31] designed a multi-satellite joint task planning and data transmission scheduling algorithm framework for multi-satellite ground observation system joint task planning and designed a task scheduling algorithm, based on task priority.

While the intelligent optimization algorithm has better solutions, the computational efficiency is lower. If the satellite task changes greatly and cannot be processed in time, the intelligent optimization algorithm cannot adapt to the requirements of in-orbit dynamic tasks. Heuristic algorithms are used to solve specific problems. While it is not guaranteed to find the best planning results, it can find a solution that meets the planning requirements in a short time. It has the advantages of a simple and easy implementation, fast and efficient solution, and intuitive and easy participation in the formulation of rules. It can be used as the basis for the algorithm design of intelligent satellite task planning.

2.2. Overview of the Application of Neural Networks in Task Planning Algorithms

The existing research, which is mainly conducted through the establishment of mathematical programming models, using various algorithms to directly solve the satellite task planning problem, has less reference to historical task planning results. Some researchers have tried to learn relevant empirical knowledge from historical task planning data using machine learning methods to solve satellite task planning problems to improve the efficiency of problem solving.

For example, Wang et al. [32] mapped the satellite task planning problem into a dynamic random knapsack problem. Asynchronous advantage actor-critic (A3C) reinforcement learning algorithm was used to determine whether each observation task should be inserted into the observation scheme in real time. Experiments show that this method is superior to the first-come-first-serve greedy algorithm in terms of optimization. Li et al. [33] combined the supervised learning method with the heuristic search algorithm and used the neural network to calculate the scheduling priority of each task. Then, the heuristic search algorithm sequentially inserts the observation task into the observation scheme, according to the scheduling priority of the task, which enhances the search ability of heuristic search algorithms. Chen et al. [34] proposed an improved genetic algorithm based on memory learning to solve the resource conflict in the evolutionary search process of a genetic algorithm by learning knowledge rules from historical planning results, which effectively improved the search efficiency and stability of the algorithm. Wang et al. [35] designed the feature expression model based on the satellite observation constraint model and the objective function, which effectively accelerated the convergence speed. Wang et al. [36] introduced the intensive learning method into the multi-satellite collaborative task assignment problem and improved the task negotiation distribution efficiency of multiple satellites by an iteratively “trial and error” search for the best negotiation strategy.

It can be seen, from the above research, that the use of machine learning methods has played a positive role in improving the efficiency and optimization of algorithms. This is also the theoretical basis of this paper.

3. Problem Description

3.1. Problem Description

During the autonomous operation of intelligent satellites, observation tasks will arrive at any time, and their sources can mainly be divided into the following three types:

1. When the iSAT satellite detects natural disasters, such as floods, mudslides, forest fires or ground targets that are covered by clouds, new observation requests are autonomously generated;
2. The iSAT satellite receives cooperative observation requests, sent by other iSAT satellites, such as the joint observation of multiple types of sensors;
3. The iSAT satellite receives a user observation request, uploaded by the ground control center or the ground user terminal.

The autonomous task planning problem for satellites is essentially a problem of maximizing the satisfaction of customer observation needs.

According to the needs of different customers, the optimization objectives have multiple manifestations, such as maximizing the observation gain, maximizing the number of observation tasks, maximizing resource utilization, etc. It may also consider multiple optimization objectives at the same time. In this paper, the commonly used maximization observation gain is used as the optimization goal, and the profit is maximized by reasonably selecting the observation task within the specified time range.

For a single iSAT satellite, we define the interval between two consecutive star-ground communication links as the entire task scheduling range. At the beginning of the scheduling range, the satellite already has several tasks to execute, and there may be new tasks to join the sequence that is to be planned. This paper does not differentiate between the two and treats them in accordance with the tasks to be planned. The total number of tasks in the two groups is N , and the two groups are collectively awaiting task planning and scheduling. The existing tasks and the newly added tasks are shown in Figure 2.

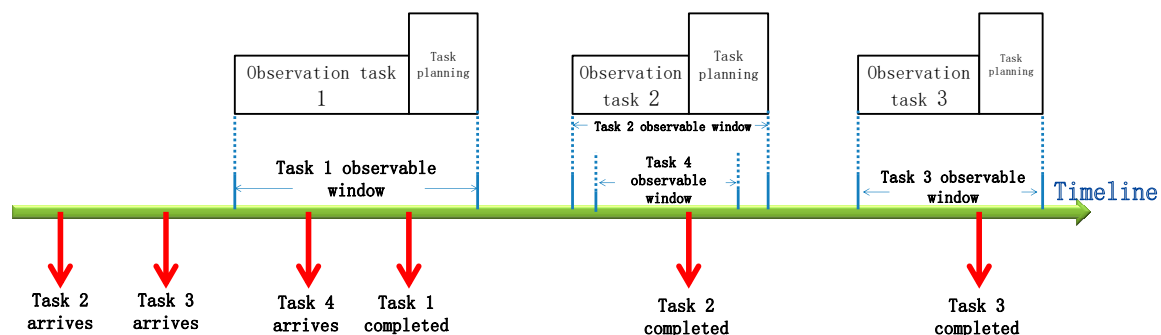


Figure 2. Schematic diagram of satellite in-orbit task planning.

3.2. Assumptions and Constraints

3.2.1. Assumptions

Based on the research of actual and realistic needs, this paper makes reasonable assumptions regarding the problems raised in this paper, which are as follows:

- (1) The scheduling range is defined as the interval between two consecutive satellite-ground communication links. In terms of time, geographically, we only plan observation tasks within a given range of observation tasks.
- (2) Intelligent satellites operate well throughout the entire dispatch range and will not be affected by space radiation effects.

- (3) Intelligent satellites have certain autonomic capabilities, allowing them to analyze the collected image information. If an event of interest is detected, a new task can be generated in-orbit.
- (4) Smart satellites can only be charged when idle (for example, without performing a task) and in the sun. The iSAT satellite can be charged while performing tasks and in the sun, but the energy obtained is much less than the energy consumed to observe the payload or transmit the payload, so the energy obtained is negligible.
- (5) When a smart satellite processes a task at any time, the task is not replaced by other tasks. That is, the observation task or the transmission task cannot be interrupted or deleted, once it is executed.
- (6) There is no priority constraint between the tasks, but each task has a constraint on the observation time window. All tasks have one and only one corresponding observation time window.

3.2.2. Constraints

Task Switching Time Constraint

Due to the strict requirements relating to the attitude of the satellite when the observation task is executed, a certain time interval must be met between two adjacent observation tasks.

Energy Constraint

Energy is a renewable resource, but the total battery capacity is limited, so the energy level cannot be higher than the upper limit of the capacity at any time, nor can it be lower than the lower limit of the capacity. The energy level drops when an observation task or a transmission task is performed, and the energy level rises while charging.

Data Storage Constraint

Storage constraints are similar to energy constraints. Due to storage capacity limitations, the total amount of data cannot be higher than the upper limit of the capacity at any time, nor can it be lower than the lower limit of the capacity. The difference is that the total amount of data increases when the observation task is performed, and the total amount of data decreases when the transmission task is executed.

4. Mixed Integer Programming Model

The essence of the satellite task planning problem is the selection of the appropriate observation task from the candidate observation tasks and determination of the execution order. Therefore, it is a typical combinatorial optimization problem. Therefore, we first express this problem as a mixed integer linear programming model.

For convenience, the symbols used in this article are as follows:

The decision variables are as follows:

$$I_i = \begin{cases} 1, & \text{if task } i \text{ is selected, } i = 0, 1, \dots, N-1 \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

$$y_{ij} = \begin{cases} 1, & \text{if task } i \text{ precedes task } j, i \neq j \\ 0, & \text{otherwise.} \end{cases} \quad (2)$$

| | |
|-----------------|---|
| <i>Indices</i> | |
| $i, j, n :$ | index |
| <i>Factors</i> | |
| $iSAT_n$ | smart satellite number |
| m_i | task number |
| $R :$ | the entire scheduling horizon |
| $Tr :$ | total length of horizon R |
| $p_i :$ | The observed profit of task i |
| $N :$ | total number of tasks in the scheduling horizon |
| $sp_{roll} :$ | rolling speed of satellite |
| $D :$ | the available date storage on the satellite |
| $d :$ | the date storage threshold on the satellite |
| $d_{initial} :$ | the initial date storage state on the satellite |
| $E :$ | the available energy on the satellite |
| $e :$ | the energy threshold on the satellite |
| $e_{initial} :$ | the initial energy on the satellite |
| $\omega :$ | the available working time on the satellite |
| $d_{i(i+1)} :$ | release date storage of task i |
| $e_{i(i+1)} :$ | release energy of task i |
| $d(T) :$ | data storage state at time T |
| $e(T) :$ | energy state at time T |
| $lf_i :$ | latest finish date of task i |
| $otw_i :$ | observation time window of task i |
| $i_i :$ | imaging time of task i |
| $\theta :$ | the rolling angle for task i |
| $s_{ij} :$ | sequence-dependent setup time if task j is executed successively after task i |
| $Ts_i :$ | observation start time of i |
| $Te_i :$ | observation end time of i |

The model is as follows:

$$\sum_{j=1, j \neq i}^N y_{ij} = I_i, \forall i = 0, 1, \dots, N-1 \quad (3)$$

$$(Ts_i + i(i))I_j + s_{ji} \leq Te_i, \forall i = 0, 1, \dots, N-1 \quad (4)$$

$$(Ts_j + i(j))I_i \leq Te_j, \forall j = 0, 1, \dots, N-1 \quad (5)$$

$$s_{ij} = \frac{|\theta_i - \theta_j|}{sp_{roll}} \quad (6)$$

$$Te_i \leq l f_i I_i, \forall i = 0, 1, \dots, N-1 \quad (7)$$

$$Ts_i + i(i) = Te_i, \forall i = 0, 1, \dots, N-1 \quad (8)$$

$$d_{initial} \sum_{i=0}^{N-1} d_i I_i \leq D \quad (9)$$

$$e \leq e_{initial} + \sum_{i=0}^{N-1} e_i \leq E \quad (10)$$

$$\sum_{i=0}^{N-1} (i_i I_i + \sum_{j=0}^{N-1} s_{ij} y_{ij}) \leq \omega \quad (11)$$

$$P = \sum_{i=0}^{N-1} p_i I_i \quad (12)$$

Equation (3) defines the rule that, if a task command arrives at the satellite and is accepted by the satellite, the task is executed only before one task and can only be completed by one task. If the above rules are not accepted, the task does not enter the task planning sequence. Receiving task j means that task j becomes a task before task i .

Equation (4) and Formula (5) show that if task i has been determined in the planning arrangement, task j is after task i , and the completion time of task j needs to consider the sequence-related setting time between task i and task j , and the imaging time of task j .

Equation (6) shows that, assuming that the observation task j is executed after the observation task i , there is a switching time s_{ij} between task j and task i .

Equation (7) shows that all tasks in the sequence are completed before the latest completion time.

Equation (8) shows that the completion time of the task is equal to the start time of the task plus the imaging time of the task.

Equations (9)–(11) indicate that the total memory, total energy, and total working time of all tasks in the planning sequence do not exceed the available memory space, total energy cap, and working time.

Equation (12) presents the total profit of the satellite during the planning process.

The core purpose of task planning is to maximize P .

5. Heuristic Search Algorithm Based on a Symmetric Recurrent Neural Network

5.1. Algorithm Design

Due to the limitations of computing resources on the star, researchers mainly use a heuristic search algorithm with a fast calculation speed to solve them. The main idea is to sort the observation tasks according to the set rules and then determine whether the task can be inserted into the observation plan. It can be seen that the optimization performance of these heuristic search algorithms largely depends on the collation designed by the researchers. Because the empirical knowledge is difficult to express, or the information is not comprehensive enough, the rule-based heuristic search algorithm is usually worse than the exact search algorithm and intelligent optimization algorithm.

In order to improve the optimization performance of the heuristic search algorithm, the paper proposed a heuristic search algorithm. The main difference between the algorithm and the existing heuristic algorithm is that the ranking rules of the observation tasks are no longer designed based on the experience of researchers, but are rather learned from the planning results of the precise search algorithm using the deep neural network algorithm, so that the ranking rules can be expressed more accurately. Specifically, the workflow of the heuristic search algorithm is shown in Figure 3:

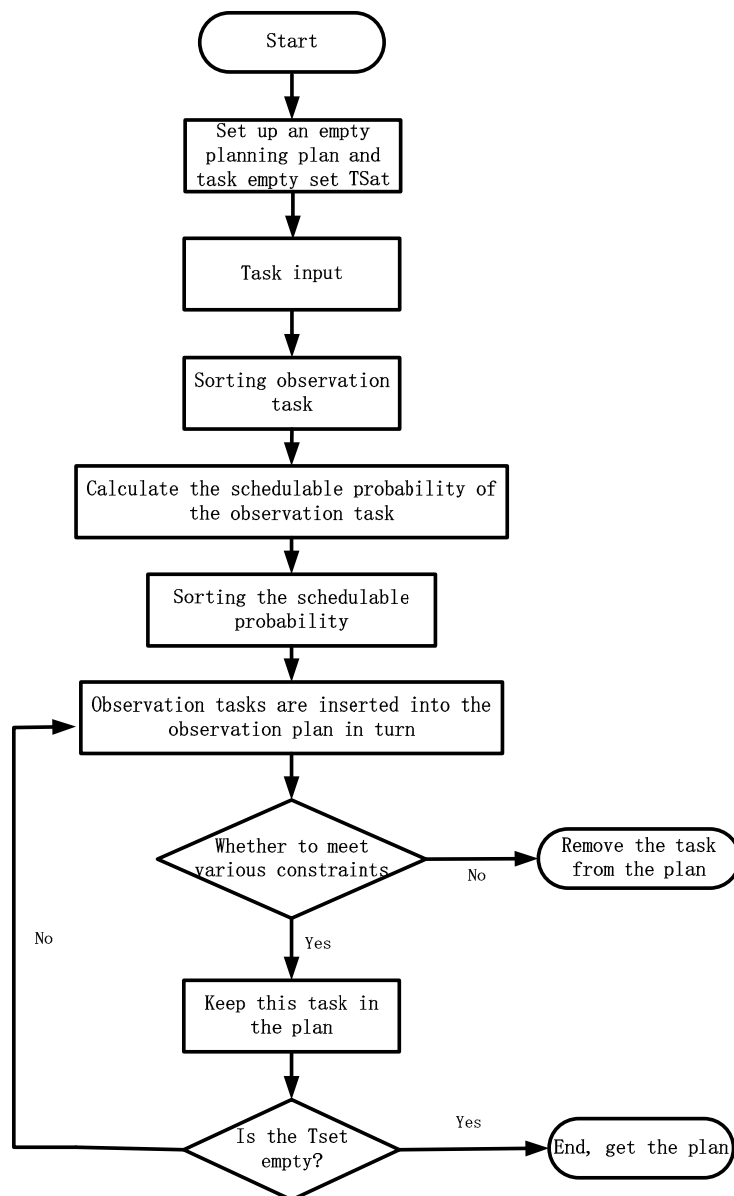


Figure 3. Algorithm flowchart.

Step 1: Select all the observation tasks in the planning period and record them as the set, TSet;

Step 2: Align the observation tasks in the set, TSet, in chronological order and use the deep neural network to sequentially calculate the schedulable probability of each observation task in the TSet (that is, the probability that the task will be executed);

Step 3: Sort the observation tasks in the TSet in descending order of schedulable probability values;

Step 4: Select the first observation task of the TSet (denoted as FTask), insert it into the observation plan, and delete the FTask from the TSet; check whether the observation plan violates resource constraints, such as energy and storage, and task switching constraints. If a constraint is violated, the task FTask is removed from the scenario.

Step 5: Repeat step 4 until the TSet is empty; then, a complete observation plan is obtained.

The Algorithm 1 shown as follow:

Algorithm 1 Heuristic Search Algorithm Based on a Symmetric Structure Neural Network

- 1: Algorithm initialization: set empty planning scheme and task empty set TSet;
- 2: Input all observation tasks within the planning into TSet;
- 3: Separate the observation tasks in the TSet in chronological order;
- 4: Calculate the schedulable probability of each observation task in the TSet;
- 5: Sort the observation tasks in the TSet;
- 6: Insert the best FTask into the observation scheme to perform constraint checking;
- 7: Repeat step 5 to output the observation plan.

In the above process, the schedulable probability of the observation task determines the ordering of the task, which is the key to affecting the performance of the heuristic algorithm. However, the historical planning result data only tell us which tasks were executed or not and does not directly give the schedulable probability of the observed task. It is not difficult to understand that the higher the schedulable probability value of the observation task, the more likely it is that the task will be executed. Then, we can use the historical planning data to train a task classification model, whose output contains two states, execution and non-execution. We can use the probability that the task will be executed as the schedulable probability of the observed task. Next, we can elaborate on the calculation method of the schedulable probability.

In order to calculate the schedulable probability of the observation task, we designed a symmetric neural network model to calculate the schedulable probability of the observation task. The network structure is shown in the following figure. In the graph, the first recurrent neural network reads the task sequence on the left side of the observation task, and the second recurrent neural network reads the task sequence on the right side of the observation task. Finally, the executed probability, P_s , and the non-execution probability, $P_n = (P_s + P_n = 1)$, of the observation task are given by the fully connected neural network; then, P_s is the schedulable probability value of the observation task. The symmetric recurrent neural network model is shown in Figure 4.

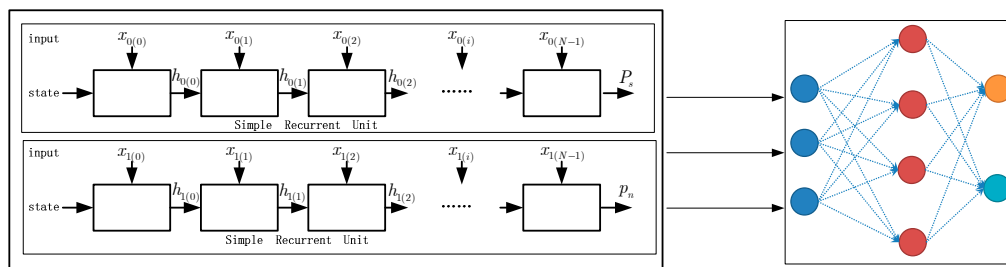


Figure 4. Schematic diagram of the symmetric recurrent neural network model.

5.2. Model Input

Using the recurrent neural network to calculate the schedulable probability of an observation task, the first question to consider is how to convert the original observation task data into input data that are recognizable by the recurrent neural network. That is, the way to represent the observation task, so that it can contain the attribute information of the observation task and the distribution relationship between the tasks. In this regard, we define the eigenvectors of the observed tasks. We take the input of the first recurrent neural network as an example. $M_{input} = \{m_0, m_1, \dots, m_{N-1}\}$ is set as an observation task sequence, arranged in chronological order, from small to large, and m_0 is the observation task to be determined. Since $\forall m_i \in M_{input}$, its eigenvector can be expressed as x_i :

$$x_i = (p_i, e_i, d_i, Te_i - Ts_i, Ts_{i+1} - Te_i, e_{(i-1)i}, d_{(i-1)i}, c_{(i-1)i}, c_{0i})$$

where p_i is the benefit value of the observation task m_i , e_i is the energy resource that the observation task m_i needs to consume, and d_i is the storage resource that the observation task m_i needs to consume;

$Te_i - Ts_i$: Observation task m_i 's continuous observation time.

$\tau_{i+1} = Ts_{i+1} - Te_i$: The time interval from observation task m_i to observation task m_{i+1} , where $\tau_0 = 0$.

$c_{i(i+1)}$: If the observation task m_i and the observation task m_{i+1} satisfy the task switching constraint, then $c_{i(i+1)} = 0$; otherwise, $c_{i(i+1)} = 1$.

$e_{(i-1)i}$: The amount of energy change between observation task m_{i-1} and observation task m_i .

$d_{(i-1)i}$: The amount of data storage change between observation task m_{i-1} and observation task m_i .

c_{0i} : If observation task m_{i-1} and observation task m_i satisfy all constraints, then $c_{0i} = 0$; otherwise, $c_{0i} = 1$.

Based on the eigenvector representation of the observation task, we constructed the original input data of the recurrent neural network, as shown in the matrix. In the figure, each column corresponds to an observation task, and each row corresponds to the characteristic value of the observation task in terms of its dimensions.

The observation task sequence Λ can be represented by eigenvector x .

$$\Lambda = [x_0, x_1, \dots, x_i, \dots, x_{n-1}]$$

$$= \begin{bmatrix} p_0 & p_1 & \dots & p_i & \dots & p_{N-1} \\ e_0 & e_1 & \dots & e_i & \dots & e_{N-1} \\ d_0 & d_1 & \dots & d_i & \dots & d_{N-1} \\ Te_0 - Ts_0 & Te_1 - Ts_1 & \dots & Te_i - Ts_i & \dots & Te_{N-1} - Ts_{N-1} \\ \tau_0 & \tau_1 & \dots & \tau_i & \dots & \tau_{N-1} \\ 0 & e_{0(1)} & \dots & e_{0(1)} & \dots & e_{N-2(N-1)} \\ 0 & d_{0(1)} & \dots & d_{0(1)} & \dots & d_{N-2(N-1)} \\ 0 & c_{0(1)} & \dots & c_{i(i+1)} & \dots & c_{N-2(N-1)} \\ 0 & c_{01} & \dots & c_{0i} & \dots & c_{0(N-1)} \end{bmatrix} \quad (13)$$

5.3. Optimization Algorithm and Loss Function

We use the cross-entropy cost function (see Formula (14)) and the Adam optimization algorithm [37] to train the schedulable probability calculation model based on the recurrent neural network.

$$L = \sum_{X \in D} - \sum_{y=1}^K p(y, X) \log P(y|X) \quad (14)$$

where $p(y, X)$ is the gold one-hot distribution of the training sample X , and $P(y|X)$ is the probability distribution function of each type.

6. Experiment

6.1. Experiment Setting

6.1.1. Track Parameter Setting

The intelligent satellite used in the simulation experiment is set as a sun-synchronous orbit satellite, and its orbital parameters are selected from the STK satellite database (see Table 1 for the specific settings). The planning period is set to 1 June 2019 00:00:00 and 1 June 2019 00:00:00, with a total of 24 h. The observation time window of the satellite to the ground target is calculated by STK. Due to the orbital height of the satellite and the angle of view of the observed load, the ground point target usually has only 1 or 2 observation time windows within 24 h. In this chapter, we assume that each observation target has only one valid observation time window, that is, each observation target corresponds to an observation task.

Table 1. Track parameter setting.

| Parameter Name | Parameter Value |
|-----------------------|-----------------|
| Semimajor axis | 7065 km |
| Semimajor axis | 0.0000923 km |
| Orbital inclination | 98.7359 |
| Right ascension point | 183.922 |
| Perigee amplitude | 114.442 km |
| Angle of approach | 354.784 |

6.1.2. Experiment Setting

In order to evaluate the performance of the proposed heuristic algorithm based on the symmetric recurrent neural network in each test case, we set up 5 planning scenarios. The number of observation tasks in each planning scene is 100–500, and the step size is 100. The target is randomly selected from the urban dataset of STK. There are 10 test cases for each planning scenario.

6.1.3. Indicator Setting

In order to evaluate the performance of an algorithm, we use three evaluation indicators: planning profit, revenue gap, and calculation time.

The planned profit represents the average of the cumulative observed returns of all test cases in the task planning algorithm in a single planning scenario. The calculation method is as follows:

$$p(A) = \frac{1}{N} \sum_{k=0}^{N-1} p_k(A) \quad (15)$$

where $p_k(A)$ represents the cumulative observation benefit of algorithm A in the k test case.

The profit gap represents the average of the ratio of the cumulative observed return difference between the task planning algorithm and the benchmark reference algorithm to the cumulative reference income of the benchmark reference algorithm. The calculation method is as follows:

$$G(A) = \frac{1}{N} \sum_{k=0}^{N-1} \frac{p_k(B) - p_k(A)}{p_k(B)} \quad (16)$$

where $p_k(A)$ represents the cumulative observation profit of algorithm A in the k test case.

The calculation time represents the average value of the calculation time of all of the test cases of the person planning algorithm in a single planning scenario (from the start of planning to the generation of the observation plan). The calculation method is as follows:

$$T(A) = \frac{1}{N} \sum_{k=0}^{N-1} t_k(A) \quad (17)$$

where $p_k(A)$ represents the calculation time of algorithm A in the k test case.

6.2. Analysis of Results

The experiment compares the planning profit and planning time of the five algorithms, CPLEX benchmark algorithm, PF, d-PSB, GBDT, and SRNN-HS (Symmetric Recurrent Neural Network Heuristic Search algorithm) in terms of the global urban datasets [38–40].

Table 2 shows the planned profit and profit gap for each algorithm, Figure 5 shows the revenue error for each algorithm, Table 3 and Figure 6 show the calculation time for each algorithm.

Table 2. Experimental result.

| Scenario | Indicator | CPLEX | PF | d-PSB | GBDT | SRNN-HS |
|----------|------------|---------|---------|---------|---------|---------|
| 100 | profit | 1577.90 | 1559.17 | 1518.50 | 1295.70 | 1540.0 |
| | profit gap | | 1.73% | 3.76% | 17.91% | 2.40% |
| 200 | profit | 2303.20 | 2126.83 | 2148.80 | 2072.90 | 2225.50 |
| | profit gap | | 7.1% | 6.76% | 10.00% | 3.36% |
| 300 | profit | 2756.30 | 2391.77 | 2551.40 | 2526.60 | 2672.10 |
| | profit gap | | 12.46% | 7.46% | 08.32% | 3.06% |
| 400 | profit | 3096.50 | 2521.13 | 2866.50 | 2732.70 | 3012.40 |
| | profit gap | | 17.87% | 7.43% | 11.73% | 2.72% |
| 500 | profit | 3378.70 | 2610.70 | 3105.80 | 2915.00 | 3262.60 |
| | profit gap | | 21.70% | 8.06% | 13.72% | 3.43% |

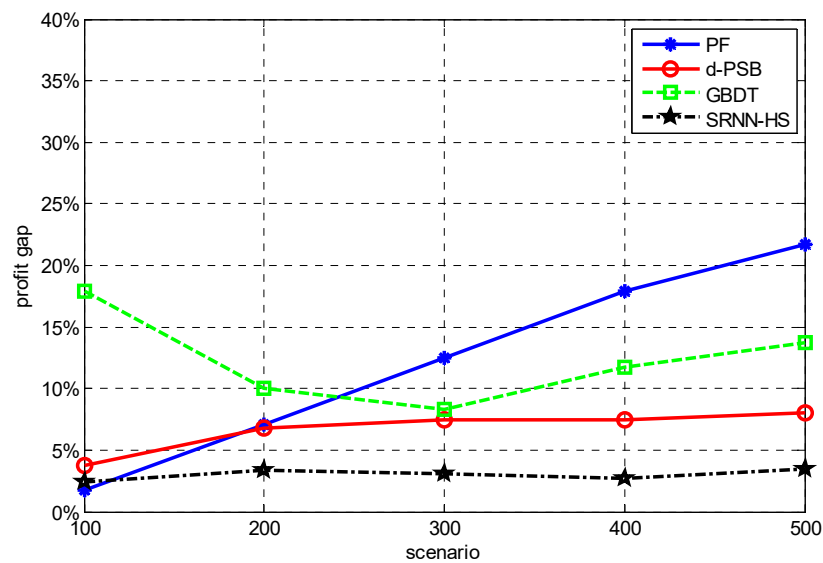


Figure 5. Profit gap comparison chart.

Table 3. Calculation time.

| Scenario | PF | d-PSB | GBDT | SRNN-HS |
|----------|-----------|-----------|-----------|-----------|
| 100 | 0.00038 s | 0.00041 s | 0.00030 s | 0.00068 s |
| 200 | 0.00047 s | 0.00058 s | 0.00030 s | 0.00077 s |
| 300 | 0.00051 s | 0.00071 s | 0.00027 s | 0.00081 s |
| 400 | 0.00055 s | 0.00079 s | 0.00024 s | 0.00085 s |
| 500 | 0.00056 s | 0.00082 s | 0.00020 s | 0.00086 s |

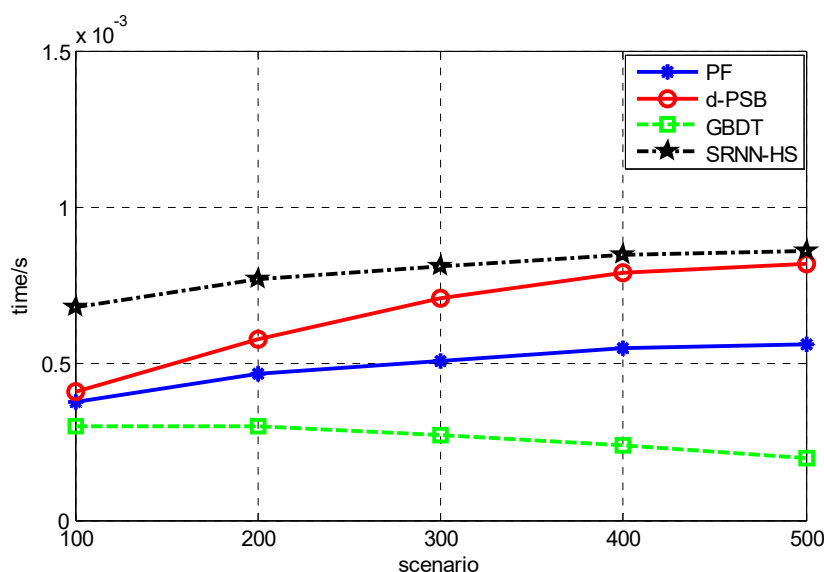


Figure 6. Time comparison chart.

The calculation result of the CPLEX benchmark algorithm is the optimal scheme for satellite task planning, but its calculation time is unacceptable for the on-board computer, so it can only be used as a benchmark algorithm for evaluating the pros and cons of the algorithm. The profit gap indicates a gap with CPLEX, and the smaller the value, the better the planning effect.

As can be seen from Table 2, the SRNN-HS proposed in this paper has achieved the highest task planning benefits in most scenarios, indicating the effectiveness of SRNN-HS.

As can be seen from Table 2 and Figure 5, in the planning scenario s100, the PF is larger in planning revenue than other algorithms, because the neural network is required to calculate the schedulable probability of each task. However, as can be seen, the calculation time increment is not very large. Even if the performance of the onboard computer is only 1/100 of that of an office computer, our algorithm can still give the planning result in a short time.

From Figure 1, it can be seen that in the planning scenario s100, PF has a less significant planning revenue gap than that of other algorithms. This is because the number of observation tasks involved in planning is lower, the degree of conflict between tasks is lower, and the selection of observation tasks with a high revenue value can achieve better results. Except for the planning scenario, s100, the SRNN-HS algorithm has less revenue error than the three heuristic algorithms, PF, d-PSB, and GBDT. This is because the task scheduling of the heuristic algorithms is obtained by researchers based on time experience. Due to the existence of blind spots in knowledge or difficulties in knowledge representation, there is still room for improvement in the optimization of rule-based heuristic search algorithms. It can be seen that SRNN-HS is also better than the GBDT algorithm. Because GBDT is a machine learning algorithm based on feature engineering, it is not comprehensive enough in the expression of features. The deep learning algorithm that we use can automatically extract features, without too much interference from researchers, and contains more effective information. The calculation cost is acceptable.

Through the above experimental analysis, the feasibility and effectiveness of the proposed algorithm are verified.

7. Conclusions

Earth observation is one of the important applications of intelligent satellites. Satellite on-orbit task planning is an important part of the intelligent satellite automatic operation. This paper describes the problem of autonomous task planning for earth observation of intelligent satellite, and establishes a mixed integer programming model for satellite mission planning. We designed a heuristic search algorithm for a symmetric recurrent neural network, based on the characteristics of the model. The schedulable probability of the task is obtained by constructing a symmetric structured recurrent neural network model. The task with the highest schedulable probability is selected to be inserted into the observation scheme, and the iterative calculation is repeated to obtain the optimal observation scheme. Considering that there is no specific research on intelligent satellite task planning, this paper compares several heuristic algorithms with several traditional observation satellite task planning algorithms. The result of experiments proved that the profit of SRNN-HS algorithm is higher than other algorithms (except s100), and the profit gap is significantly lower than other algorithms. The experimental results illustrated that the performance of SRNN-HS algorithm is better than the rule-based heuristic algorithm and the feature extraction-based machine learning heuristic algorithm. However, this algorithm brings the increase of time consumption, which is also the limitation of this algorithm. This paper presents a task planning algorithm which can be used in the background of intelligent satellite iSAT. In the future, we can consider the application of the algorithm and multi satellite cooperative autonomous task planning, so as to improve the observation range and the efficiency of earth observation.

Author Contributions: Conceptualization, S.L.; methodology, J.Y.; validation, S.L.; investigation, S.L.; writing—original draft preparation, S.L.; writing—review and editing, J.Y.; visualization, S.L.; supervision, J.Y.; project administration, J.Y.; funding acquisition, J.Y.

Funding: This research received no external funding.

Acknowledgments: The authors are grateful to the reviewers and editors for their valuable feedback on our work, which helped to improve the quality of this paper.

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

References

1. Min, S. China's intelligent space-based integrated information network concept. In Proceedings of the 2018 Software Definition Satellite Summit, Beijing, China, 4 April 2018.
2. Zhao, J.; Wu, F.; Liu, G. Development and Prospect of software definition satellite technology. *Satell. Netw.* **2017**, *12*, 46–50.
3. Ulversoy, T. Software Defined Radio: Challenges and Opportunities. *IEEE Commun. Surv. Tutor.* **2010**, *12*, 531–550. [[CrossRef](#)]
4. NASA. Space Telecommunication Radio System STRS Architecture Standard. Revision 1.02.1. 2010.
5. Reinhart, R.C. Space Communication and Navigation SDR Testbed, Overview and Opportunity for Experiments. In Proceedings of the SDR WinnComm Technical Conference, Washington, DC, USA, 8 January 2013.
6. Feng, Y. The fully softwareized “quantum” satellite is of great significance. *Def. Perspect.* **2015**, *9*, 17–18.
7. Lockheed Martin's First Smart Satellites Are Tiny with Big Missions. Available online: <https://news.lockheedmartin.com> (accessed on 20 March 2019).
8. Yang, X. Integrated Integrated Spatial Information System Based on “Software Star”. *J. China Acad. Electron. Inf. Technol.* **2004**, *5*, 15–22.
9. Li, L. Software-defined satellite intelligence leads the space flight. *China Sci. Technol. Awards* **2019**, *235*, 26–28.
10. Yang, Z. *Research on Low-Cost Digital Signal Processing Technology of Space-Based Network Intelligent Satellite*; National University of Defense Technology: Changsha, China, 2016.
11. Qin, H. *Research on Key Technologies of Intelligent Satellite (iSAT) for Space-Based Network*; National University of Defense Technology: Changsha, China, 2017.

12. Jin, P.; Yu, K. Simulation Study on Integrated Scheduling of Satellite Target Resources. *Comput. Simul.* **2018**, *35*, 16–21.
13. Berger, J.; Giasson, E.; Florea, M.; Harb, M.; Teske, A.; Petriu, E.; Abielmona, R.; Falcon, R.; Lo, N. A Graph-based Genetic Algorithm to Solve the Virtual Constellation Multi-Satellite Collection Scheduling Problem. *IEEE Congr. Evolut. Comput.* **2018**, 1–8. [[CrossRef](#)]
14. Mouw, C.B.; Greb, S.; Aurin, D.; DiGiacomo, P.M.; Lee, Z.; Twardowski, M.; Binding, C.; Hu, C.; Ma, R.; Moore, T.; et al. Aquatic Color Radiometry Remote Sensing of Coastal and Inland Waters: Challenges and Recommendations for Future Satellite Mission. *Remote Sens. Environ.* **2015**, *160*, 15–30. [[CrossRef](#)]
15. Xhafa, F.; Sun, J.; Barolli, A.; Biberaj, A.; Barolli, L. Genetic Algorithms for Satellite Scheduling Problems. *Mob. Inf. Syst.* **2012**, *8*, 351–377. [[CrossRef](#)]
16. Sarkheyli, A.; Bagheri, A.; Ghorbani-Vaghei, B.; Askari-Moghadam, R. Using an Effective Tabu Search in Interactive Resources Scheduling Problem for Leo Satellites Missions. *Aerosp. Sci. Technol.* **2013**, *29*, 287–295. [[CrossRef](#)]
17. Qi, M.R. Using Constraint Satisfaction to Cooperate Satellites' Activities for the Mission of Area Target Observation. *J. Astronaut.* **2007**, *28*, 238–242.
18. Globus, A.; Wu, G.; Liu, J. A Two-phase Scheduling Method with the Consideration of Task Clustering for Earth Observing Satellites. *Comput. Op. Res.* **2013**, *40*, 1884–1894.
19. He, R.; Gao, P.; Bai, B. Models, algorithms and applications to the task planning system of imaging satellites. *Syst. Eng. Theor. Pract.* **2011**, *31*, 411–422.
20. Gao, P.; Tan, Y.J.; Li, J.F.; He, R.J. An Ant Colony Algorithm for Remote Satellite and Groud Integration Scheduling Problem in Parallel Environment. *Adv. Mater. Res.* **2013**, *791*, 1341–1346. [[CrossRef](#)]
21. Jiang, W.; Pang, X. The group scheduling method for the tasks of imaging satellite network. *Syst. Eng. Theor. Pract.* **2014**, *34*, 2154–2162.
22. Zheng, Z.; Guo, J.; Gill, E. Onboard autonomous mission re-planning for multi-satellite system. *Acta Astronaut.* **2018**, *145*, 28–43. [[CrossRef](#)]
23. Niu, X. Dynamically Scheduling of Multiple Satellites for Response to Natural Disasters Using Genetic Algorithms. *Geogr. Geo-Inf. Sci.* **2018**, *34*, 126–130.
24. 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](#)]
25. Bina, C.; Straub, J.; Marsh, R. Scheduling Algorithm Development for an Open Source Software Spacecraft. In *University of North Dakota School of Graduate Studies Scholarly Forum*; University of North Dakota: North Dakota, ND, USA, 2015; pp. 212–223.
26. Tangpattanakul, P.; Jozefowicz, N.; Lopez, P.A. A multi-objective local search heuristic for scheduling Earth observations taken by an agile satellite. *Eur. J. Op. Res.* **2015**, *245*, 542–554. [[CrossRef](#)]
27. Chen, Y.; Sun, K. Multi-image Satellites Cooperative Observing Scheduling on GRASP. *Control Eng. China* **2012**, *19*, 691–694, 698.
28. Xue, J. *Research on Autonomous Task Planning of Earth Observation Satellite*; Nanjing Aerospace University: Nanjing, China, 2015; pp. 47–68.
29. Si, X. *Autonomous Task Planning Method of Spacecraft Based on The Constraint Satisfaction*; Beijing Institute of Technology: Beijing, China, 2015; pp. 34–58.
30. Liu, S.; Chen, Y.; Xing, L.; Sun, K. Method of agile imaging satellites autonomous task planning. *Comput. Integr. Manuf. Syst.* **2016**, *22*, 928–934.
31. Wang, Z.; Wang, Y.I. New Multi-satellite Scheduling Method. *Chin. Space Sci. Technol.* **2012**, *32*, 8–14.
32. Haijiao, W.A.; Zhen, Y.A.; Wugen, Z.H.; Dalin, L.I. Online scheduling of image satellites based on neural network and deep reinforcement learning. *Chin. J. Aeronaut.* **2019**, *32*, 1011–1019.
33. Li, C.; Chen, Y. Date-Driven Onboard Scheduling for an Autonomous Observation Satellite. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, Beijing, China, 7 July 2018; pp. 5773–5774.
34. Chen, H.; Zhou, Y.; Du, C.; Li, J. A Satellite Cluster Date Transmission Scheduling Method based on Genetic Algorithm with Rote Learning Operator. In *Proceedings of the IEEE Congress on Evolutionary Computation*, Vancouver, BC, Canada, 24–29 July 2016; pp. 5076–5083.
35. Wang, C.; Chen, H.; Zhai, B.; Li, J.; Chen, L. Satellite Observing Mission Scheduling Method base on Case-Based Learning and a Genetic Algorithm. In *Proceedings of the IEEE International Conference on Tools with Artificial Intelligence*, San Jose, CA, USA, 6–8 November 2017; pp. 627–634.

36. Wang, C.; Jing, N.; Li, J.; Wang, J.; Chen, H. An Algorithm of Cooperative Multiple Satellites Task planning Based on Multi-agent Reinforcement Learning. *J. Univ. Def. Technol.* **2011**, *33*, 53–58.
37. Kingma, D.; Ba, J. Adam: A Method for Stochastic Optimization. *Comput. Sci.* **2014**, 27–43.
38. Li, G.; Xing, L.; Chen, Y. A hybrid online scheduling mechanism with revision and progressive techniques for autonomous Earth observation satellite. *Acta Astronaut.* **2017**, *140*, 308–321. [[CrossRef](#)]
39. Li, G. *Models and Algorithms for Distributed Earth Observation Satellite System Online Coordination Scheduling under Communication Constraints*; National University of Defense Technology: Changsha, China, 2017.
40. Haykin, S. *Neural Networks and Machine Learning*; Mechanical Industry Press: Beijing, China, 2011.



© 2019 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 (<http://creativecommons.org/licenses/by/4.0/>).