mathematics

Article

Dynamic Network Resource Autonomy Management and Task Scheduling Method

Xiuhong Li [1], Jiale Yang [1] and Huilong Fan [2,*]

[1] College of Information Science and Engineering (School of Cyber Science and Engineering), XinJiang University, Urmuqi 830046, China
[2] School of Computer Science and Engineering, Central South University, Changsha 410075, China
* Correspondence: hlfanpro@csu.edu.cn

Abstract: Satellite network resource management and scheduling technology are significant to constructing integrated information networks in heaven and earth. The difficulty in realizing this technology lies in improving resource utilization efficiency while ensuring the service quality of satellites and efficiently coordinating complex satellite network systems and services. This paper proposes a model, A Dynamic task scheduling method based on a UNified resource Management architecture(DUNM), based on the designed resource management architecture supported by dynamic scheduling algorithms to address the problems of low resource utilization, resource allocation, and task completion rate. First, with sufficient resources, the task execution time to complete a task is calculated based on the number of resources, task transmission time, task waiting time, etc. Secondly, based on the tasks assigned to satellites, the execution time of all functions with different transmission rates of communication links between satellites is calculated, and the total sum of all time consumption is analyzed. Finally, after simulation experiments and comparison with various baseline algorithms, about a 40% reduction in time to complete scheduled tasks and an almost 25% reduction in the average cost to finish a scheduling task, our method has higher scheduling efficiency and lower task completion revenue. It also guarantees a higher task completion rate while completing the tasks. Our approach attained a nearly 100% completion rate for scheduling tasks, which means that our algorithm can achieve the scheduling tasks faster and at high task revenue, thus improving the efficiency and economic efficiency of the whole system. Therefore, it validates the advantages of our method, such as high efficiency and high revenue.

Keywords: spatial networks; dynamic networks; task scheduling; resources management

MSC: 68M18

Citation: Li, X.; Yang, J.; Fan, H. Dynamic Network Resource Autonomy Management and Task Scheduling Method. *Mathematics* **2023**, *11*, 1232. https://doi.org/10.3390/math11051232

Academic Editors: José F. Vicent, Leandro Tortosa and Manuel Curado

Received: 3 February 2023
Revised: 27 February 2023
Accepted: 1 March 2023
Published: 3 March 2023

1. Introduction

Satellite network resource management and scheduling technology are significant to constructing integrated information networks in heaven and earth [1]. This technology is a scientific method that can efficiently manage and schedule various satellite resources, such as power, bandwidth, time, etc., to meet the relevant needs of satellite networks. Ensure the effectiveness and reliability of satellite networks, improve the quality of satellite services, and ensure the effective execution of the tasks assigned to satellites [2]. Resource management refers to managing resources in the satellite network to improve the utilization of resources [3]. The scheduling process of this technology has the following steps: 1. Receive task requests, receive and process service requests from various types of satellites, determine the resource demand of the requested task and the corresponding time requirements, etc. 2. Predict the resource situation of the satellite network. 3. Determine the resource allocation plan of the requested task based on the resource demand of the job assigned to the satellite and the situation of the resources available in the satellite network.

4. Task scheduling for the already allocated resources, determining the transmission of tasks according to different requirements, etc. 5. Starting the execution of satellite tasks according to the defined planning, performing communication, and ensuring the orderly execution of tasks. 6. Monitoring the indicators of the satellite network. 7. Analyzing the scheduling results and determining the next step [4].

Due to the increasing complexity of satellite networks, this technology has many challenges in the research process. In the execution of satellite scheduling tasks, the satellite network resource management and scheduling technology also have some difficulties in the process of implementation, which include the following aspects: 1. The significant number of satellites leads to the problem of managing resources in the sky orbit [5]. 2. Because of the resource demand for satellites, The need for a dynamic scheduling method is even more critical because of the rapid changes in satellite resource requirements [1]. 3. When performing the assigned tasks, it is essential to achieve fair scheduling due to the more diverse service needs to be met by the satellite network. 4. During the operation of the satellite network, there may be scheduling disturbances due to the operating environment's complexity. As the complexity of satellite computing continues to increase, finding an efficient algorithm to implement the scheduling task is essential. Furthermore, Satellite network resource management and scheduling technology is a relatively popular research direction. With the research of scholars and institutions in China and abroad, some initial progress has been made, such as the satellite scheduling method based on a genetic algorithm and a popular research direction in satellite network resource management and scheduling technology.

Some general satellite scheduling methods include dynamic scheduling methods based on resource demand and fairness-based scheduling algorithms. The research mainly focuses on balancing each satellite's market and available resources. Still, the process also has shortcomings, such as insufficient anti-interference ability and difficulty evaluating the fairness index. Hence, future research needs to be carried out more deeply in future research [6], solutions for interference, etc.

Furthermore, in modern communication, 5G, IoT, remote sensing, and other fields, satellite scheduling, and satellite systems also play an irreplaceable role, and it is increasingly vital to ensure the efficiency and reliability of the scheduling system. In addition, since the traditional scheduling algorithm cannot adapt well to the dynamic change problem in the satellite scheduling process, we must improve and optimize the old algorithm using a dynamic scheduling algorithm. Finally, based on the resource management architecture, we can better manage and utilize the resources, thus improving the system's efficiency.

Therefore, finding a more comprehensive and efficient dynamic scheduling algorithm to solve the problems arising in the satellite scheduling process is increasingly important. For the unsolved problems, we combine the scheduling ideas of resource evaluation, mission scheduling, mission execution, mission feedback, and evaluation by summarizing the previous studies. Our approach is to monitor the usage of satellite resources in real-time through unified management and configuration of satellite resources with the support of dynamic scheduling algorithms. The resources are reasonably allocated according to the priority of satellite missions and available resources to ensure efficient utilization. Last, we make sure the efficient operation of satellites and complete tasks.

The contributions of this paper are summarized as follows. 1. To solve the under-utilization of resources, we include resource sharing in our method to achieve dynamic allocation and uniform distribution of resources and improve the resource utilization of the satellite network. 2. To address the problem of efficient task execution, we ensure the high timeliness of the tasks assigned in the satellite network through resource monitoring technology and the monitoring of satellite status so that they can be completed within the specified time, increasing the task completion rate. 3. For the scheduling fairness problem, our proposed dynamic scheduling algorithm is more reasonable and fair in the satellite scheduling process than other algorithms. 4. Solving the problem that traditional scheduling algorithms are difficult to adapt to the dynamic changes in the process of satellite schedule. 5. Through simulation experiments, we verify the advantages of our proposed method, such as efficiency and effectiveness, and conclude that the technique can improve task completion efficiency, optimize resource utilization, and cope with changing task demands, thus enhancing the efficiency of the whole system.

Compared with other satellite scheduling-related work, the main feature of our work is the use of dynamic scheduling algorithms based on resource management architecture for optimization. On the one hand, it can better grasp the resource situation of satellites, and on the other hand can make adaptive adjustments for the system, thus ensuring the system's reliability and improving the scheduling efficiency. Our innovative and practical work can provide relevant references for research and applications in this field.

2. Related Work

With the continuous development of satellite applications [3], the satellite industry is also undergoing a revolution, and in the context of current 5G network applications [3,7,8], how to provide faster data transmission rates, low-latency services, etc. [9–12], and how to efficiently and effectively It is particularly important to plan, coordinate, and execute satellite scheduling and tasks under limited resources [13–15], and to maximize the efficiency of resource use while ensuring that all tasks are met [2,10,16–28]. For the problems in traditional satellite network scheduling, such as limited satellite resources, coordination between multi-level scheduling, satellite interference, and how to effectively handle the interaction between satellites and networks, these problems will also lead to satellite resource utilization [29–31]. In scheduling satellites, the imbalance between scheduling and information transmission will cause the network performance to be not very high [20,32]; some parameters are not Meets expectations, such as transfer delays between networks and the rate at which messages are delivered [11].

Aiming at the insufficient resource utilization and application problems in satellite task scheduling, Yang et al. proposed a software-defined satellite network architecture (SDSN). The architecture is designed to simplify networking between multifunctional satellites and provide higher resource utilization. SDSN uses a new protocol to enable testing and deployment to achieve this goal. After testing, the seamless handover mechanism of SDSN has significantly improved handover delay and throughput. However, although SDSN supports flexible low-level protocols, it fails to adapt to some high-level protocols [7,33,34]. This means that SDSN may be unable to meet some specific applications' needs. Therefore, in future development, SDSN may need to provide more protocol adaptation functions to meet a broader range of application scenarios.

Zhou et al. proposed a method for achieving fair performance among user satellites, a mixed-integer nonlinear program optimization problem. To solve this problem, they decompose it into two equivalent subproblems. Among them, the task scheduling problem of optimal power allocation is also a mixed integer linear programming problem. Experimental results show that this algorithm has certain advantages in solving task scheduling problems [35,36]. Although this algorithm performs well in some cases, it may face convergence difficulties when dealing with large-scale problems.

SoftSpace, proposed by Xu et al., is a new generation of satellite network software-defined architecture, which adopts concepts such as network function virtualization and aims to reduce capital and operating costs and improve the interoperability of satellite network equipment [2,37,38]. However, although SoftSpace excels in many ways, it has some challenges, such as security issues and controller selection issues that have not yet been resolved. These issues are the critical points that SoftSpace needs to pay attention to in future development.

Zhou et al. proposed a throughput-maximizing mixed-integer linear programming approach to solve the problem of low throughput and joint management between energy and transponder resources in small satellite networks. The method uses a scalable time-evolution graph to describe network resources and experiments with high throughput on a time-slot basis. Simulation results show that this method can achieve high throughput with low complexity [36]. However, although this method performs well in ensuring efficient use of the network, it does not consider the priority among multiple tasks and the reliability issues among the various links of satellite scheduling [39].

The cross-domain SDN architecture proposed by Shi et al. can make upgrading and configuring the underlying foundation more accessible, and the architecture's performance is verified in practical cases. The results show that this architecture can effectively reduce the control time overhead of configuration updates and decisions [8,40]. However, although the cross-domain SDN architecture is excellent in many aspects, it ignores the development of interface protocols for the interconnection between logical planes. Currently, there is no specification for reference in this regard, so how to formulate a suitable interface protocol according to different links is still a problem to be explored [41].

To solve the dynamic management problem among multiple resources in the satellite information network, Wang et al. considered the random observation and transmission channel conditions in the satellite information network during the research process and established a comprehensive optimization framework. To ensure the effectiveness of the network, they formulated a traffic optimization problem based on time-expanded graphs. They proposed an online algorithm using the Lyapunov optimization technique theory to solve the traffic optimization problem on time-expanded graphs in a slot-by-slot manner [1,42]. However, this approach performs well in solving the problem of multiple resource management in satellite information networks. However, it does not effectively consider the latency problem of switching between multiple observation targets when observing satellites. It also fails to integrate the different effectively adaptable quality of service parameters required. These issues are the critical points that the method needs to pay attention to in future development.

Therefore, in response to the above-unresolved issues, we design an adaptive management architecture for resource unification and allocation and propose an improved dynamic scheduling algorithm adapted to dynamic satellite network changes to solve these problems.

3. Dynamic Resource Management Architecture and Task Scheduling Method

To address the problems of efficient utilization of satellite network resources, diversification of satellite network task demands, and management and scheduling of satellite network resources, we first design the management architecture of dynamic resources to realize dynamic allocation and unified distribution of resources. Then we develop the scheduling algorithm based on this architecture and finally learn the efficient utilization of resources and improve the high timeliness of scheduling.

As shown in Figure 1, this is the overall framework diagram of our study.

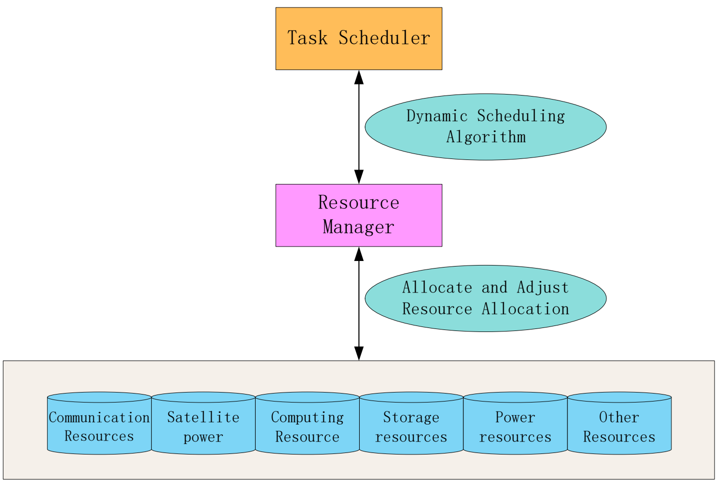

Figure 1. Overall framework diagram.

3.1. Dynamic Resource Management Architecture

Traditional satellite scheduling is a systematic process used to manage satellite operations and monitor the status of the satellite, etc. First, the ground station will acquire data on the satellite's position through communication with the satellite. Then, the ground station will analyze the collected data to determine the satellite's current status, which will help the assigned mission determine the satellite's following action. The ground station will develop a corresponding mission plan based on the satellite's current status and the task to be performed, which will guide the satellite's operation in the future. The ground station sends mission plan commands to the satellite using a communication link with the satellite. After the mission is finished, the ground station again collects data from the satellite for analysis to ensure the satellite's mission performance, assess the need for mission adjustments, and maintain the satellite's condition. Finally, to ensure that the satellite can efficiently and reliably complete the assigned mission, ensuring the satellite's integrity and stability. Finally, it ensures that the satellite can efficiently and reliably achieve its assigned task, ensuring the integrity and stability of the satellite and that it provides high-quality services over the long term.

Compared with traditional satellite scheduling, satellite scheduling with local network can improve the efficiency of information interaction and data transmission between satellites, enhance the ability to monitor satellites in real time, optimize satellite resources in real time, so as to improve efficiency and make better use of satellite resources; it can respond quickly to changes in environmental conditions, so that satellites can adapt to some new situations and can Optimize the performance of the satellite, improve the efficiency of satellite fault diagnosis and maintenance; can also be based on the current state of the satellite and the mission needs of the dynamic allocation of the required resources to improve the utilization of resources and improve the overall efficiency of the satellite; can also be faster to respond to some satellite failure problems and so on to reallocate resources, enhance the reliability and fault tolerance of the satellite scheduling process; and for some It can also support some tasks with high complexity and distribution.

Our proposed resource management architecture, as shown in Figure 2, has four parts: resource monitoring, resource forecasting, resource scheduling, and resource allocation. In this architecture, as shown in Figure 3, there are four primary layers: the resource layer, monitoring layer, virtual layer, and application layer. We can propose a scheduling algorithm to satisfy the task while making the best possible use of the satellite's resources. Based on our proposed resource management framework, we offer a scheduling algorithm to solve the problem of calculating the sum of the time consumed by the satellite during the execution of all tasks in the context of different transmission rates of the communication links between satellites.

In resource monitoring, we collect various resource usage of satellites through multiple sensors and monitoring systems, such as the power of satellites, communication bandwidth, satellites' computing power, etc. We define the total amount of satellite resources as R_{s_j}, we record all satellites as set $S = \{s_1, s_2, \cdots, s_n\},\ n \geq 1,\ n \in \mathbb{N}^*$, and each satellite represents $s_j,\ j \in \{1, 2, \cdots, n\}$. In real-time, relevant data can be collected and transmitted back to the ground control center (such as the network operation control center), which will be used for subsequent resource forecasting and scheduling. In the resource prediction section, we will also use the monitored data to predict the future use of the satellite's various resources and can make timely adjustments accordingly. By analyzing the historical data and the previously collected ground information, we can derive the trend of each resource usage of the satellite. We will adjust the resource allocation strategy of the satellite according to the predicted resource usage to avoid a shortage of resources and underutilization of resources.

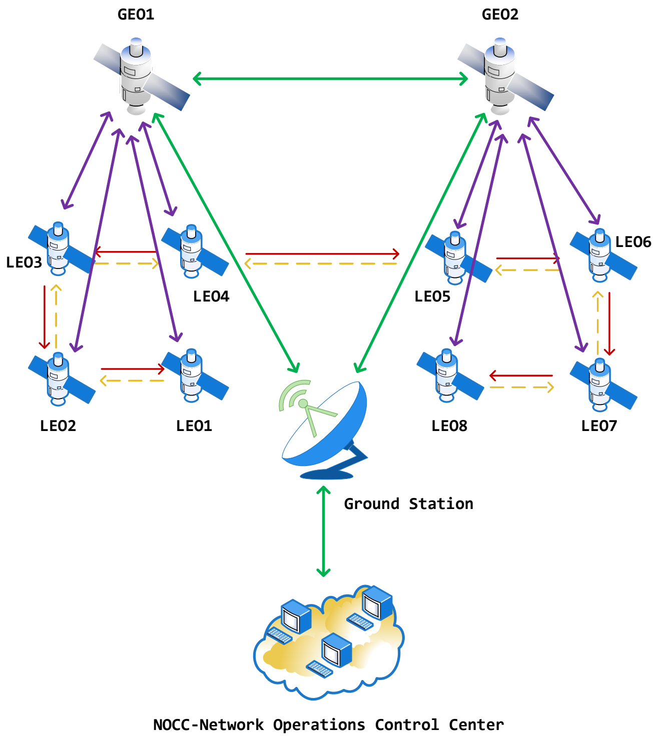

Figure 2. Satellite network resource management architecture diagram graph.

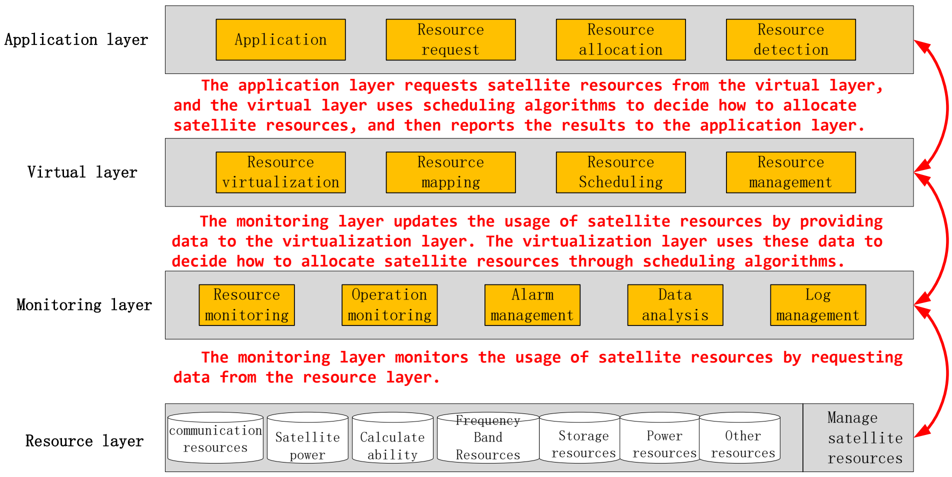

Figure 3. Hierarchical structure diagram.

In the resource scheduling section, we arrange the resource allocation of the satellite according to the usage of various satellite resources and the priority and time requirements of each task distributed. We denote the number of resources required by the satellite to execute a specific task as $r(\tau)$; we suggest all the lessons be completed by the satellite as set $\Gamma = \{\tau_1, \tau_2, \cdots, \tau_m\},\ m \geq 1,\ m \in \mathbb{N}^*$ and each task is marked as$\gamma_i,\ i \in \{1, 2, \cdots, m\}$. When the functions of the satellite are waiting for scheduling or execution, it will generate the corresponding waiting time $W(\tau_u) = \sum_{\tau \in \xi} E(\tau)$. $E(\tau)$ denotes the task schedule execution time, and these data are constant. If the waiting time is too long, it may delay the satellite tasks and affect the user experience. We record the task time in transmission to and back from the satellite as the task transmission time $\psi(\tau)$. As shown in Figure 4. At the moment ϱ, the change matrix after task scheduling is $M_\varrho = M_\Gamma(\varrho) - M_\Gamma(0)$. This time can reflect the transmission rate in the satellite system. If the transmission time process is, it will significantly reduce the system efficiency and affect the user experience. In this part of resource scheduling, the system will plan the various resource allocations of the satellite according to the current task requirements and resource usage to meet the needs of multiple tasks and maximize resource usage efficiency, thus improving the efficiency of the system.

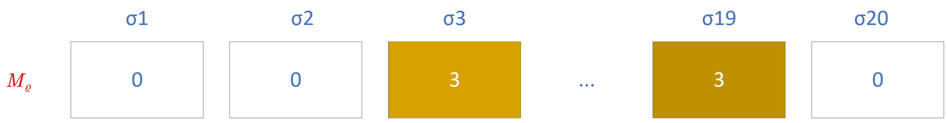

Figure 4. The difference between the satellite number of the task at a certain moment and the satellite number of the task at the initial time, if it is 0, it means that the task has not been scheduled to other satellites, if it is not 0, it means that the task has been scheduled to other satellites.

In the resource allocation section, we assign satellite resources to each mission to meet each mission's requirements. Our proposed framework has extensive conditions for mission execution as long as the satellite has enough resources to execute. In this part, the system monitors the progress of each task in real-time. It adjusts the system's allocation of satellite resources to ensure that the tasks assigned by the system can be completed within the specified time. When some tasks are completed, the number of resources corresponding to the satellite will be released. We denote the number of released resources by $R^f_{s_j} = \sum_{\tau\in\xi} r(\tau)$, where ξ, $\xi \in \Gamma'_{s_j}$ represents the set of completed tasks, and $r(\tau)$, $\tau \in \xi$ denotes the number of resources required for task τ. The remaining amount of resources for the satellite is $R^l_{s_j} = R_{s_j} - R^f_{s_j}$.

The resource layer's primary role is to monitor and manage the satellite's various resources, including the satellite's power, communication broadband, and computing power. The resource providers of the resource layer are mainly various satellites, flight equipment, etc. The resource layer is also responsible for managing the satellite resources, allocating and releasing resources according to the missions, and ensuring that the satellite resources are efficiently allocated to all assignments. The results of the resource layer can be provided to the next layer, and the monitoring and management results of the resource layer can provide the necessary data for the monitoring layer. The resource layer can also manage the resources after the virtualization of the virtual layer.

The monitoring layer monitors the satellites' operational status, performance, and related parameters to ensure proper operation. The monitoring layer also collects a lot of operational data from the satellite pairs and provides it to the next layer for analysis. Accordingly, the results of the monitoring layer can help the resource layer better monitor and manage the satellites' various resources. Based on the results of the monitoring layer, the resource layer can also do proper planning with the help of the missions assigned to the satellites. The monitoring layer performs registration, capability modeling, semantic acquisition, and semantic analysis on the resources acquired by the resource.

The primary role of the virtual layer is to support the virtualization of resources, which can realize the unified management and share among satellite resources, thus improving the efficiency of satellite resource utilization. According to the results of the monitoring layer, the virtual layer virtualizes the resources and generates virtual resources. The coating contains the generation of virtual resources of satellites, the modeling of some responsible mission requirements, etc. The virtual layer also constructs the semantic model through the results of the monitoring layer. The virtual layer can improve the efficiency of satellite resources by virtualizing them and providing the results to the application layer for analysis.

The primary role of the application layer is to provide services to the relevant users. The application layer can realize the interaction between users and the satellite system, so this layer is an interface between the satellite system and the users. The application layer forwards the service requests received from the users to the corresponding parts for processing and use. The application layer can also use satellite-related resources through the virtual layer and monitor various satellite operational statuses and other data through the monitoring layer. The application layer's prominent role is to manage the virtual resources and query whether the satellite resources are updated to process the data. The application layer can also monitor the status of the satellites through the monitoring layer to ensure the proper operation of the satellites. The application layer needs to meet the various needs of the users in the process of use. It should also consider the efficiency of utilizing the satellite resources and the priority of the tasks assigned to the satellite.

3.2. Dynamic Graph Construction Based on Local Satellite Networks

We implement the architecture through a series of functional designs, which can effectively coordinate the relationship between the transmission rates of satellites, navigation satellites, and communication links between satellites and make it easy to collect the status of satellite network resources and the operational status of each satellite. Our proposed framework can reflect the realistic satellite system more realistically. It can consider the impact of the limitation of communication broadband between satellites on the total time consumption. By calculating the total time consumption, we can determine what needs to be optimized to improve the efficiency and performance of the system, which is something we need to improve in the future.

In controlling the enslaved person, as shown in Figure 5, the master can transmit commands and information, such as data and execution links. The enslaved person, through direct communication between satellites or with the assistance of ground stations, will then receive commands from the master and execute them; the master will also ensure the proper execution of the enslaved person by periodically monitoring the status of the enslaved person. Suppose there are other problems, such as anomalies in one enslaved person. In that case, the master will coordinate the entire system to ensure If one of the enslaved people has abnormalities or other problems, the master will coordinate the whole system to ensure the proper operation of the satellite. The master can also assign tasks to enslaved people based on their capabilities and status by evaluating their qualifications, including their processing and task execution capabilities. The enslaved person receives and executes the tasks assigned by the master. In addition, the master star completes the satellite scheduling based on the state of the slave star under the strategy of the dynamic scheduling algorithm.

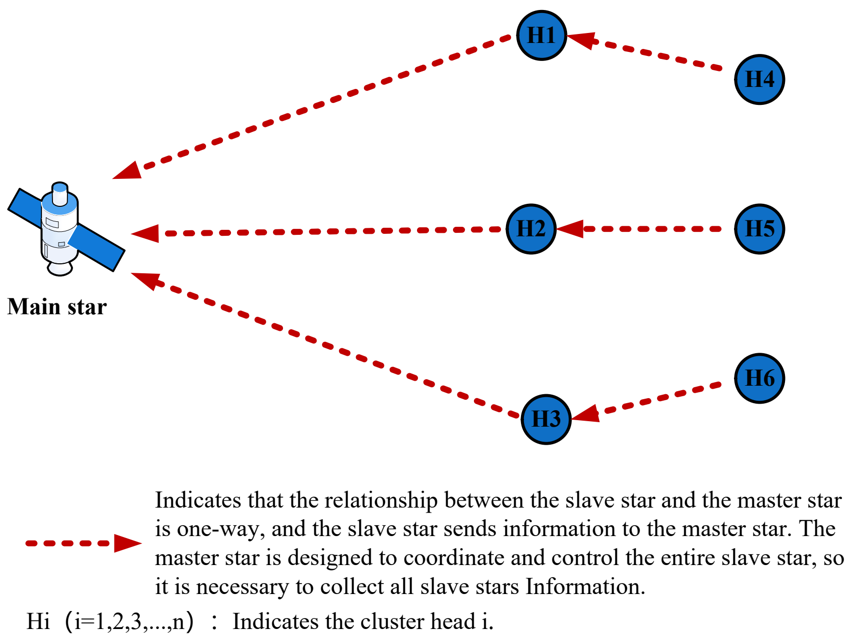

Figure 5. Master-slave relationship diagram.

The master star can control the cluster head to maintain the normal state of the whole cluster. The master star can send control commands to the cluster head periodically to update the cluster state information, use the link to monitor the cluster's operational state, and adjust the position and speed of the slave star through the cluster head.

As shown in Figure 6, the cluster head can control the cluster members through command and data links. For example, it can send control commands to the members, etc. The cluster head and members also communicate and collaborate through specific protocols. The cluster head is responsible for organizing and managing the cluster members, assigning tasks, and coordinating resources. Cluster members are responsible for receiving and executing tasks and reporting the progress of jobs to the cluster head. The interaction between cluster heads and cluster members is the key to collaborating to accomplish the satellite scheduling tasks together.

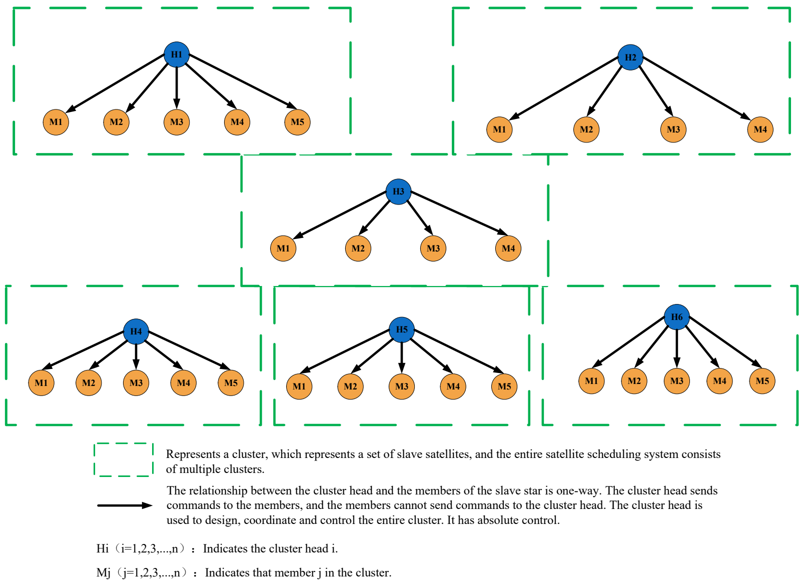

Figure 6. Inter-cluster relationship diagram.

3.3. Satellite Mission Scheduling

In our proposed framework, we denote the set of all tasks to be executed by a satellite as a task queue. The tasks about the satellite may be requested by the user or assigned according to the satellite system. The tasks in the task queue will be arranged according to a certain priority, which we denote as set $\Gamma_{s_j}^{'} = \{\tau_1, \tau_2, \cdots, \tau_p\},\ p \geq 1,\ p \in \mathbb{N}^*$, which is also the input of our proposed model, and a certain task on a satellite is noted as $\tau_k^{'},\ k \in \{1, 2, \cdots, p\}$. For the output of our model, we specify that if task $\tau_i \in \Gamma_{s_j}^{'}$, then there exists $\tau_i = \tau_k^{'}$, marking the state of task τ_i on satellite s_j as $M_{ij} = 1$; otherwise, mark it as $M_{ij} = 0$. We define a mission-affiliated satellite matrix, $M_{\Gamma}^{t} = \begin{bmatrix} \sigma_1^t & \sigma_2^t & \cdots & \sigma_h^t \end{bmatrix},\ \sigma_i \in \{1, 2, \cdots, n\},\ h = p,\ t \in \delta$, where δ denotes the discrete period, and the mission matrix element is the satellite number indicating the satellite number to which the mission belongs.

As shown in Figure 7, when $t = 0$, $M_{\Gamma}(0)$ is the initial state of the satellite to which the mission belongs.

When t=ϱ, $M_{\Gamma}(\varrho)$ is the initial state of the satellite belonging to the task at the moment ϱ. (It should be noted that only one satellite is allowed to perform a mission)

The matrix we have defined represents the satellites to which each character belongs and their status. This matrix helps to determine to which satellite the task should be assigned and allows us to make an optimal decision to choose which satellite is suitable for which task.

Regarding the objective of our proposed algorithm, we denote the execution time of a particular task $\tau_u,\ u \in \{1, 2, \cdots, m\}$ as $T(\tau_u)$, and we have.

Task execution time $T(\tau_u)$ = task scheduled execution time $E(\tau_u)$ + task waiting time $W(\tau_u)$ + task transmission time $\psi(\tau)$.

From this, we can calculate the execution time of all tasks:

$$
\begin{aligned}
T &= \sum_{\tau \in \Gamma} T(\tau) \\
&= \sum_{\tau \in \Gamma} E(\tau) + W(\tau) + \psi(\tau) \\
&= \begin{cases} \sum_{\tau \in \Gamma} \left(E(\tau) + \sum_{\{\gamma:\gamma \in \xi,\ \xi \in \Gamma\}} E(\gamma) + \sum_{\{k:1 \le k \le h\}} Tran\left(s_{\sigma_k^0}, s_{\sigma_k^\varrho} \right) \right) & M_\varrho \neq 0 \\ \sum_{\tau \in \Gamma} \left(E(\tau) + \sum_{\{\gamma:\gamma \in \xi,\ \xi \in \Gamma\}} E(\gamma) \right) & M_\varrho = 0 \end{cases} \\
&.t. \\
&(\tau_u) \le R^l_{s_j}, \\
&\sigma_k \in M_\varrho.
\end{aligned}
$$

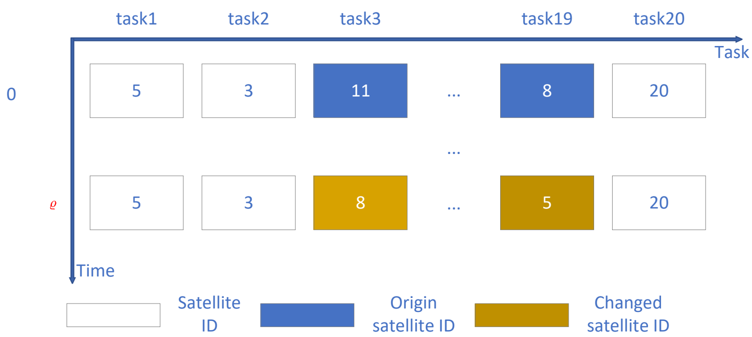

Figure 7. The matrix representation of the tasks on the satellite, the horizontal axis represents the tasks that increase in order, the vertical axis represents the time, and the value in the element is the satellite number.

According to Algorithm 1, we first sort the tasks according to their priority and deadline (line 1) and select the task with the highest priority and the closest deadline for scheduling. If the task is feasible, it proceeds to schedule the task (lines 2–3), and if the task is not achievable, it updates its priority based on the new information and puts it back into the ready queue (lines 4–5). The algorithm continues until all tasks have been scheduled or the ready string is empty.

Algorithm 1: Highest Priority Scheduling

Input: Task set *tasks*.
Output: Schedule of tasks.
1 Sort *tasks* based on their priority and deadline; *ready_queue* ← *tasks*; **while** *ready_queue is not empty* **do**
2 *task* ← get next task from *ready_queue*; **if** *task.is_feasible*() **then**
3 Schedule *task*;
4 **else**
5 Update task priority based on new information; Put the task back to *ready_queue*;

4. Simulation Experiments

We analyze the performance, advantages, and disadvantages of the method in this paper by comparing many simulation experiments with the baseline method. The experimental simulation data mainly include synthetic data based on actual satellite-related data and simulation data generated based on satellite and mission characteristics. The results show that our proposed algorithm has the advantages of high efficiency and high task revenue in accomplishing the scheduling tasks. As shown in Table 1, the following table describes the operating environment of the experiment.

Table 1. Introduction to running environment configuration.

Configuration Items	Computer Configuration Information
Operating System	Windows 10 Family Chinese Version
CPU	Intel(R) Core(TM) i5-1035G1 CPU @ 1.00 GHz
Memory	8 GB
Hard Disk	512 GB
Video Cards	NVIDIA GeForce MX350

4.1. Simulation Platform and Data

To study the critical problems of mission scheduling, link optimization, collaborative computing, and network topology discovery of spatial information networks, we simulated satellite data, including satellite load data, satellite orbit data, inter-satellite visible time window, and satellite ground visible time window data, satellite resource capacity, and satellite resource quantity. In addition, we simulated data of common Earth observation application scenarios, including Earth observation area, mission execution time demand, mission resource type, mission resource demand, etc. The system created a multi-satellite and multi-task earth observation scene. The scene start time is "13 June 2021 04:00:00.000 UTCG", and the end time is "14 Jun 2021 04:00:00.000 UTCG". The three-dimensional and two-dimensional views of the satellite and the earth in the Earth observation simulation scene are shown in Figure 8. As shown in Figure 8, we depict the track of the subsatellite point in the two-dimensional plane. We denote the satellite coordinates in its orbital plane as cr=(x_0, y_0, z_0). The calculation formula (1) is as follows,

$$cr = \begin{cases} x_0 = -a\left(1-e^2\right) \times \frac{\cos(\theta)}{1+e\cos(\theta)} \\ y_0 = -a\left(1-e^2\right) \times \frac{\sin(\theta)}{1+e\cos(\theta)} \\ z_0 = 0 \end{cases} \tag{1}$$

where a denotes semi major axis, θ indicates trueanomaly, e represents eccentricity.

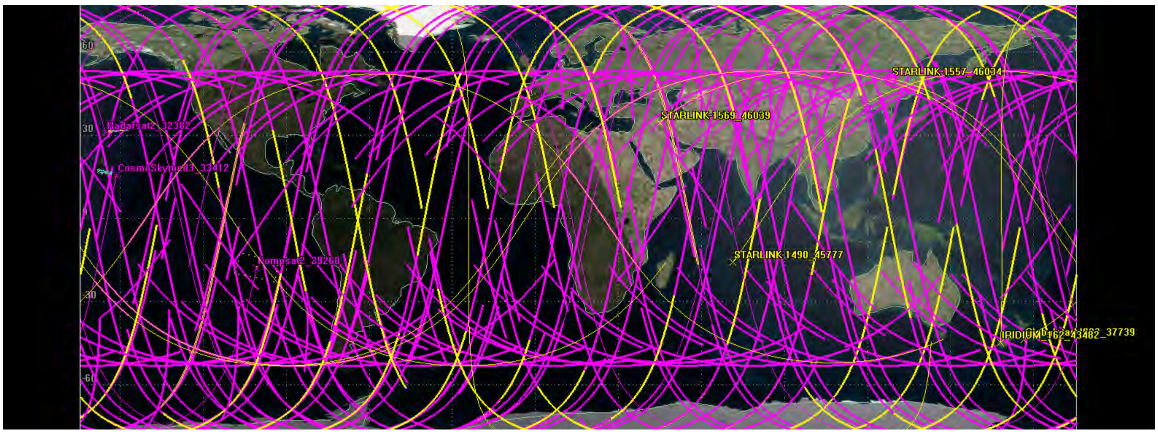

Figure 8. Two-dimensional trajectory of the satellite.

4.2. Numerical Comparative Analysis

In our study, we define the evaluation metric of the algorithm as the task revenue [43], and a larger value of this value indicates a better algorithm effect. The value is calculated as

$$M(X_{sub}) \triangleq T(t)/p_i(t)$$

The algorithm for comparison with our proposed algorithm is as follows: Machine learning methods include DS (Dual Simplex Algorithm, DS) [44], PS (Primal Simplex Algorithm) [45], NS (Network Simplex Algorithm) [46], Barrier(Barrier Algorithm) [47] and Sifting(Sifting Algorithm) [48], all of which are based on the power of the solver CPLEX.

In Table 2, in the running time comparison results table between different algorithms, the horizontal axis indicates various satellite scheduling algorithms, and the vertical axis shows the task dataset. The vertical axis from top to bottom suggests increasing the task dataset, i.e., data volume. The data in the table indicates the time various algorithms take to complete the task. It can be seen that when the data set is of the same standard, our proposed DUNM algorithm has a significant improvement in the index of time to complete the scheduling task compared with other machines' learning-based PS [45], DS [44], NS, Barrier [47], and Sifting algorithms, especially when the data set is smaller, e.g.,

Table 2. The running time comparison results between DUNM and the other five algorithms.

	DUNM	**PS**	**DS**	**NS**	**Barrier**	**Sifting**
D1	0.465873	0.813586	0.841538	0.796589	0.802961	0.774817
D2	0.591948	0.901514	0.865037	0.838193	0.851823	0.881752
D3	0.733071	0.878968	0.892657	0.94071	0.904292	0.895728
D4	0.893934	0.972274	0.899088	0.970249	0.898302	0.968244

When the data set is D1, the DUNM running time is. For example, when the data set is D1, the DUNM running time is 0.465873, PS is 0.813586, DS is 0.841538, etc. Our proposed algorithm reduces the time to complete the scheduling task by nearly 40% compared with other algorithms. As the data set increases, our proposed algorithm reduces the time to complete the scheduling task compared with different algorithms. For example, in data sets D2, D3, and D4, the time taken by DUNM to complete the scheduling task is less than the rest of the algorithms. As shown in Figure 9, according to the line graph, we can more intuitively find that our proposed algorithm is shorter in running time as the dataset grows.

The reason for this gap is that, on the one hand, since these algorithms are based on machine learning, a lot of time is needed for model training before testing the models. There is also data to process to generate more task spend and overhead. For example, in the case of the DS algorithm, if the amount of data for the satellite schedule is large, the algorithm may need more time to process it. The Primal-Simplex (PS) algorithm is similar to the DS algorithm because it is a mathematical optimization algorithm with high computational complexity, so it may take more time to complete the scheduling task. The remaining algorithms, such as the NS algorithm, Barrier algorithm/citezhang2018breach, and Sifting algorithm, are similar to the above algorithms because they have high computational complexity. They are less efficient compared to our proposed algorithm, resulting in a longer time required to complete the task.

While machine learning-based algorithms usually need to train the model first and then perform a lot of computation when predicting the task, our approach relies more on real-time data and some state information of the system. It does not need to train the model in advance. Linear programming-based algorithms require longer computation time in large-scale linear programming problems and may be affected by the data size. Our proposed algorithm focuses more on the dynamic allocation and scheduling of resources and is suitable for task-scheduling problems in dynamic scenarios. Therefore, our proposed algorithm is not too high in terms of computational and algorithmic complexity, and the algorithm is also simple and computationally efficient, which can process data quickly and complete the satellite scheduling task. Secondly, our proposed algorithm is highly flexible. It can adapt to the dynamic changes of the satellite scheduling task, thus shortening the time to complete the task and improving the system's efficiency.

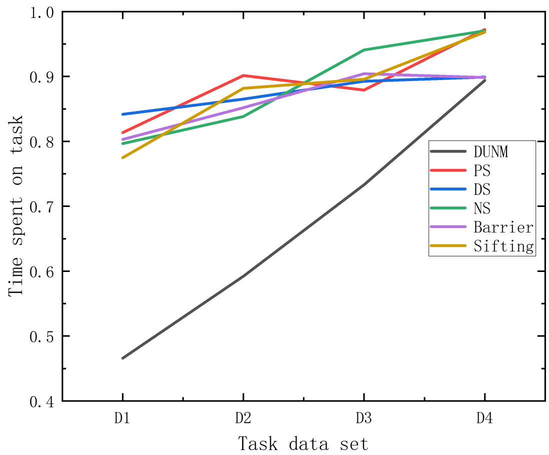

Figure 9. Comparison of the algorithm's running times for different data sets.

As Table 3, in the data table of task revenue comparison results corresponding to various algorithms, the first column indicates the data sets DI, D2, D3, and D4 and the increasing data sets from top to bottom. The second column indicates various algorithms under different stages of data sets, such as the DUNM algorithm, PS algorithm [45], DS algorithm [44], NS algorithm [46], Barrier algorithm [47], and Sifting algorithm [48] proposed in this study. The third and fifth columns indicate the metrics of the various algorithms at different stages of the dataset: the amount of successfully assigned tasks, the amount of unsuccessfully assigned tasks, and the total revenue, respectively.

Table 3. Comparison of the total revenue of completing scheduling tasks between DUNM and the other five algorithms.

Data Set	Algorithm	Task Successfully Assigned	Failed to Assign Task	Total Revenue
D1	DUNM	42	0	51.53
	PS	40	2	38.52
	DS	40	2	38.52
	NS	40	2	38.52
	Barrier	40	2	38.52
	Sifting	40	2	38.52
D2	DUNM	77	0	156.12
	PS	65	12	82.7
	DS	65	12	82.7
	NS	65	12	82.7
	Barrier	65	12	82.7
	Sifting	65	12	82.7
D3	DUNM	102	0	161.63
	PS	46	56	125.7
	DS	46	56	125.7
	NS	44	58	123.07
	Barrier	46	56	125.7
	Sifting	46	56	125.7
D4	DUNM	149	0	298.67
	PS	62	87	121.4
	DS	62	87	121.4
	NS	62	87	121.4
	Barrier	63	86	121
	Sifting	61	88	120.95

At different stages of the task dataset, our proposed algorithms successfully assigned all the scheduling tasks to the satellites, e.g., at dataset D1, all the tasks successfully assigned by our proposed algorithms amounted to 42, and the total revenue was 51.53; at dataset D2, the tasks successfully assigned amounted to 77, and the total revenue was 156.12; at dataset D3, the tasks successfully assigned amounted to is 102, generating a total revenue of 161.63; at data set D4, the number of successfully assigned tasks is 149, generating a total revenue of 298.67; all the remaining algorithms compared increase the number of unsuccessfully assigned tasks as the data set continues to increase, and even when the task data set is larger, the number of successfully assigned tasks is lower than the number of unsuccessfully assigned tasks; for example, the PS algorithm, at data set D1, the amount of successfully assigned tasks is 40, and the total revenue is 38.52; in data set D2, the amount of successfully assigned tasks is 65, and the total revenue is 82.7; in data set D3, the amount of successfully assigned tasks is 46, and the total revenue is 125.7; in data set D4, the amount of successfully assigned tasks is 62, and the total revenue is 121.4. Comparing the revenue of completing a task, the revenue of DUNM is 1.23, 2.03, 1.58, and 2.00 in data sets D1, D2, D3, and D4, respectively; the PS algorithm is 0.963, 1.27, 2.73 and 1.96, respectively; the rest of the algorithms are similar to PS, and all have lower task revenue. All of our proposed algorithms have higher total revenue for the dataset scheduling task because we complete more scheduling tasks, so we spend more. By comparing the task revenue of completing a task, our algorithm is significantly more effective than other algorithms, and the more the number of tasks in the dataset, the more effective our algorithm becomes. This is sufficient to show the efficiency of our proposed algorithm.

The probability of successfully assigned and unsuccessful tasks, which is an important indicator of the effectiveness of the scheduling algorithm, is 100% for DUNM. In contrast, the probability of successfully assigned tasks decreases with the dataset increase for the rest of the algorithms. Even when the dataset is D4, the probability of successfully assigned tasks for the algorithm PS is less than 42%, so it can be concluded that DUNM is more effective in the process of satellite schedule. In contrast, the rest of the algorithms are less effective. The rest of the algorithms are less effective. This result is because the dynamic scheduling algorithm can be adjusted in real-time according to the task requirements during the execution of the satellite scheduling task to improve resource utilization and scheduling efficiency and can cope better when the dataset is larger. As shown in Figure 10, according to the line graph, we can more intuitively find that our proposed algorithm increases the number of successfully assigned tasks as the dataset continues to grow, and the number of unsuccessfully assigned tasks has been 0. At the same time, the revenue per task is also low.

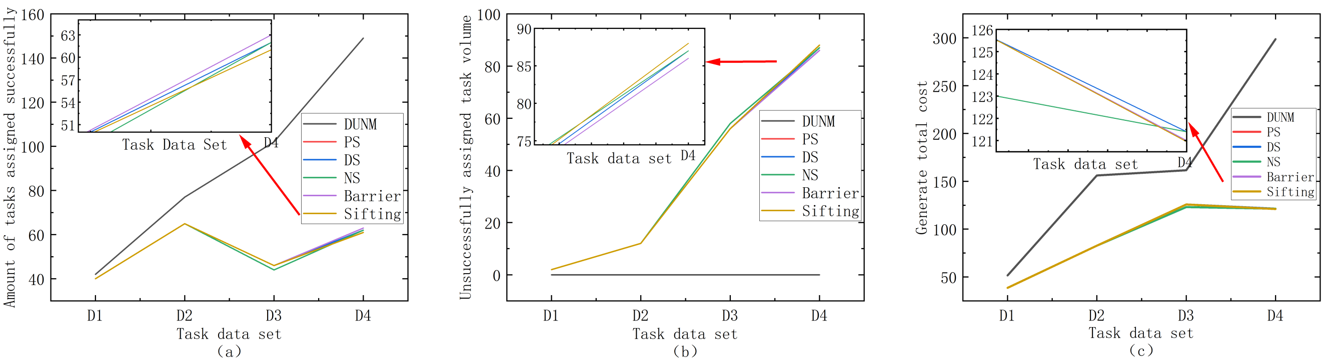

Figure 10. Performance comparison of various algorithms on different task datasets. (**a**) Successful task assignments. (**b**) Unsuccessful task assignments. (**c**) Total revenue comparison.

As in Table 4, in the task revenue comparison results data table, the table describes the data on the number of tasks that have been assigned, the number of unsuccessfully assigned tasks, and the task revenue by the six algorithms for different task cycles.DUNM, PS [45], DS [44], NS [46], Barrier [47], and Sifting [48] represent the six different satellite scheduling algorithms. Each task has data corresponding to the assigned tasks in each mission, and each algorithm has data such as the assigned tasks, the unassigned tasks, and the task revenue. For example, in the T1 task, algorithm SUNM has eight assigned tasks and 0 unassigned tasks, and the revenue of these eight tasks is 4.42; similarly, for the other tasks, the assignment of algorithm SUNM and the corresponding revenue can be observed. The same is true for the other algorithms, and this table shows their allocations in the tasks and the corresponding revenue, respectively.

Table 4. Comparison results of DUNM and the other five algorithms in the task revenue of completing scheduling tasks.

		T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10
DUNM	assigned tasks	8	16	21	31	4	6	10	0	3	7	9	30	4	7	8	15	4	6	8	12	2	7	7	11	3	6	11	13	5	6	9	12	4	8	9	11	5	8	10	14
	unassigned tasks	0	0	0	0	0	0	0	18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Task revenue	4.42	28.57	34.08	56.32	4.18	19.2	24.47	0	9.86	16.6	7.96	54.7	0.73	9.62	14.41	30.95	12.59	16.37	16.24	33.65	1.04	25.14	8.41	26.1	4.52	9.52	25.66	29.3	6.73	18.61	5.34	25.23	4.16	6.05	10.79	26.06	3.3	6.45	14.27	16.36
PS	assigned tasks	4	6	4	6	4	7	4	6	6	8	6	9	5	7	5	8	3	5	4	5	3	6	4	5	3	8	6	8	4	5	4	4	4	7	4	6	4	6	5	5
	unassigned tasks	4	10	17	25	4	9	23	37	1	8	26	40	0	8	29	47	1	9	33	54	0	10	36	60	0	8	41	65	1	9	46	73	1	10	51	78	2	12	56	87
	Task revenue	3.76	10.73	16	14.5	2.68	9.93	19.67	13.17	9.51	10.42	14	11.58	2.77	5.9	21.29	17.58	3.57	8.77	13	9.33	1.31	8.45	5.92	7.73	2.48	6.24	14.5	13.33	4.36	10.93	5.67	12	3.59	5.35	8.67	12.67	4.5	5.98	7	9.5
DS	assigned tasks	4	6	4	6	4	7	4	6	6	8	6	9	5	7	5	8	3	5	4	5	3	6	4	5	3	8	6	8	4	5	4	4	4	7	4	6	4	6	5	5
	unassigned tasks	4	10	17	25	4	9	23	37	1	8	26	40	0	8	29	47	1	9	33	54	0	10	36	60	0	8	41	65	1	9	46	73	1	10	51	78	2	12	56	87
	Task revenue	3.76	10.73	16	14.5	2.68	9.93	19.67	13.17	9.51	10.42	14	11.58	2.77	5.9	21.29	17.58	3.57	8.77	13	9.33	1.31	8.45	5.92	7.73	2.48	6.24	14.5	13.33	4.36	10.93	5.67	12	3.59	5.35	8.67	12.67	4.5	5.98	7	9.5
NS	assigned tasks	4	6	4	6	3	7	4	6	6	8	4	9	5	7	5	8	3	5	4	5	4	6	4	5	3	8	6	8	4	5	4	4	4	7	5	6	4	6	4	5
	unassigned tasks	4	10	17	25	5	9	23	37	2	8	28	40	1	8	31	47	2	9	35	54	0	10	38	60	0	8	43	65	1	9	48	73	1	10	52	78	2	12	58	87
	Task revenue	3.76	10.73	16	14.5	2.01	9.93	19.67	13.17	9.57	10.42	11	11.58	2.77	5.9	21.29	17.58	3.57	8.77	13	9.33	1.91	8.45	6.45	7.73	2.48	6.24	14.5	13.33	4.36	10.93	5.67	12	3.59	5.35	8.67	12.67	4.5	5.98	6.83	9.5
Barrier	assigned tasks	4	6	4	6	4	7	4	6	6	8	6	9	5	7	5	8	3	5	4	5	3	6	4	5	3	8	6	8	4	5	4	4	4	7	4	6	4	6	5	6
	unassigned tasks	4	10	17	25	4	9	23	37	1	8	26	40	0	8	29	47	1	9	33	54	0	10	36	60	0	8	41	65	1	9	46	73	1	10	51	78	2	12	56	86
	Task revenue	3.76	10.73	16	14.5	2.68	9.93	19.67	13.17	9.51	10.42	14	11.58	2.77	5.9	21.29	17.58	3.57	8.77	13	9.33	1.31	8.45	5.92	7.73	2.48	6.24	14.5	13.33	4.36	10.93	5.67	12	3.59	5.35	8.67	12.67	4.5	5.98	7	9.1
Sifting	assigned tasks	4	6	4	6	4	7	4	6	6	8	6	8	5	7	5	8	3	5	4	5	3	6	4	6	3	8	6	7	4	5	4	4	4	7	4	6	4	6	5	5
	unassigned tasks	4	10	17	25	4	9	23	37	1	8	26	41	0	8	29	48	1	9	33	55	0	10	36	60	0	8	41	66	1	9	46	74	1	10	51	79	2	12	56	88
	Task revenue	3.76	10.73	16	13.5	2.68	9.93	19.67	13.5	9.51	10.42	14	11.75	2.77	5.9	21.29	17.58	3.57	8.77	13	9.33	1.31	8.45	5.92	7.78	2.48	6.24	14.5	13.33	4.36	10.93	5.67	12	3.59	5.35	8.67	12.67	4.5	5.98	7	9.5

At task T1, our proposed DUNM algorithm has successfully allocated all the tasks, 8, and the rest of the algorithms have failed to allocate half of the tasks, 4. It is more obvious when there are many tasks. For example, at task T4, our proposed algorithm has successfully allocated 31 tasks, and the rest of the algorithms have allocated six tasks, which is less efficient in comparison.

Regarding the task revenue for completing the scheduling tasks, our proposed DUNM Sofa, although it appears larger in the table, is because our algorithm completes more scheduling tasks. For example, in scheduling task T4, algorithm DUNM completes 31 tasks with a task gain of 56.32. PS algorithm [45] completes six tasks with a task gain of 14.5. Other algorithms are similar. In comparison, our algorithm completes scheduling tasks with higher gains, sufficient to show the high gain of our proposed algorithm advantage.

Because of the higher efficiency of our proposed dynamic scheduling algorithm, it can have faster decision-making capabilities in complex situations and under multi-task conditions. In addition, the method can be adapted to the constraints of satellite resources. Multiple objectives, such as task completion time and resource usage efficiency, can be considered simultaneously, thus ensuring the efficiency and fairness of task allocation and satisfying scheduling requirements.

The remaining algorithms, such as machine learning-based scheduling algorithms, require a large amount of data for training, which can be time-consuming and may also suffer from low training efficiency.

The above experimental procedure mathematically involves mathematical optimization problems and some algorithms. In our proposed approach, network resources are considered finite and allocatable, and tasks are entities that need to use these resources. Therefore, the scheduling problem between tasks and resources can be abstracted as an optimization problem, i.e., how to allocate tasks with limited resources to achieve scheduling purposes. Beyond that, mathematical modeling and algorithms are the core of solving optimization problems. Mathematical models such as linear programming can be used to represent task and resource constraints and specify optimal scheduling policies. Thus, our research is very closely related to mathematics.

5. Conclusions

This study proposes a dynamic task scheduling method based on a unified resource management architecture, which has obvious advantages in satellite schedule. The technique combines resource management with dynamic scheduling algorithms, realizes real-time monitoring of resources, can better adapt to the satellite scheduling environment, and can intelligently allocate tasks according to their priority and available resources. Experiments show that the method can guarantee the completion rate of satellite scheduling tasks and improve resource utilization. The results of this study provide the necessary guidance for the design and management of satellite scheduling algorithms and systems and verify the effectiveness and practicality of the method. Importantly, our research is relevant regarding sustainability, market demand, and impact on policy. Our approach can reduce energy consumption by improving resource utilization, reducing environmental impact, and meeting changing market demands flexibly with dynamic scheduling algorithms. It helps related companies improve productivity and competitiveness, as well as can help realize digital transformation and informatization to promote global economic prosperity and development. However, our architecture still needs to be improved. For example, more resource types, task types, and performance metrics can be introduced to more comprehensively consider the utilization of resources, meet different requirements, and evaluate the overall system's performance. Where machine learning and deep learning algorithms can provide strong support for improving such architectures, there is more significant potential for development in this area. In the future, we will continue to delve into extending the approach to other applications.

Author Contributions: Conceptualization, H.F.; Methodology, X.L., J.Y. and H.F.; Software, J.Y. and H.F.; Validation, X.L. and J.Y.; Formal analysis, J.Y. and H.F.; Investigation, X.L., J.Y. and H.F.; Resources, H.F.; Data curation, H.F.; Writing—original draft, H.F.; Writing—review and editing, X.L., J.Y. and H.F.; Visualization, X.L.; Project administration, X.L. and H.F.; Funding acquisition, X.L.

Funding: This research was funded by Xinjiang Natural Science Foundation grant number 2020D01C026.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Wang, Y.; Sheng, M.; Ye, Q.; Zhang, S.; Zhuang, W.; Li, J. Optimal Dynamic Multi-Resource Management in Earth Observation Oriented Space Information Networks. *arXiv* **2019**, arXiv:1907.12717.
2. Sheng, M.; Wang, Y.; Li, J.; Liu, R.; Zhou, D.; He, L. Toward a flexible and reconfigurable broadband satellite network: Resource management architecture and strategies. *IEEE Wirel. Commun.* **2017**, *24*, 127–133. [CrossRef]
3. Ferrús, R.; Koumaras, H.; Sallent, O.; Agapiou, G.; Rasheed, T.; Kourtis, M.A.; Boustie, C.; Gélard, P.; Ahmed, T. SDN/NFV-enabled satellite communications networks: Opportunities, scenarios and challenges. *Phys. Commun.* **2016**, *18*, 95–112. [CrossRef]
4. Wu, H.; Chen, J.; Zhou, C.; Shi, W.; Cheng, N.; Xu, W.; Zhuang, W.; Shen, X.S. Resource management in space-air-ground integrated vehicular networks: SDN control and AI algorithm design. *IEEE Wirel. Commun.* **2020**, *27*, 52–60. [CrossRef]
5. Miao, Y.; Cheng, Z.; Li, W.; Ma, H.; Liu, X.; Cui, Z. Software defined integrated satellite-terrestrial network: A survey. In Proceedings of the International Conference on Space Information Network, Yinchuan, China, 10–11 August 2017 ; Springer: Berlin/Heidelberg, Germany, 2017; pp. 16–25.
6. Xie, G.; Liu, L.; Yang, L.; Li, R. Scheduling trade-off of dynamic multiple parallel workflows on heterogeneous distributed computing systems. *Concurr. Comput. Pract. Exp.* **2017**, *29*, e3782. [CrossRef]
7. Yang, B.; Wu, Y.; Chu, X.; Song, G. Seamless handover in software-defined satellite networking. *IEEE Commun. Lett.* **2016**, *20*, 1768–1771. [CrossRef]
8. Shi, Y.; Cao, Y.; Liu, J.; Kato, N. A cross-domain SDN architecture for multi-layered space-terrestrial integrated networks. *IEEE Netw.* **2019**, *33*, 29–35. [CrossRef]
9. Li, T.; Zhou, H.; Luo, H.; Yu, S. SERvICE: A software defined framework for integrated space-terrestrial satellite communication. *IEEE Trans. Mob. Comput.* **2017**, *17*, 703–716. [CrossRef]
10. Du, J.; Jiang, C.; Qian, Y.; Han, Z.; Ren, Y. Resource allocation with video traffic prediction in cloud-based space systems. *IEEE Trans. Multimed.* **2016**, *18*, 820–830. [CrossRef]
11. El Sabbagh, E.M.; Hussein, M.; Elbayoumy, A.D. Enhancing the End-to-end Link Performance of Traditional Satellite Ground Networks via Software-Defined Networking Controllers (POX/RYU). *Int. J. Comput. Appl.* **2020**, *975*, 8887.
12. Guo, Q.; Gu, R.; Dong, T.; Yin, J.; Liu, Z.; Bai, L.; Ji, Y. SDN-based end-to-end fragment-aware routing for elastic data flows in LEO satellite-terrestrial network. *IEEE Access* **2018**, *7*, 396–410. [CrossRef]
13. Liu, R.; Sheng, M.; Lui, K.S.; Wang, X.; Wang, Y.; Zhou, D. An analytical framework for resource-limited small satellite networks. *IEEE Commun. Lett.* **2015**, *20*, 388–391. [CrossRef]
14. Bao, J.; Zhao, B.; Yu, W.; Feng, Z.; Wu, C.; Gong, Z. OpenSAN: A software-defined satellite network architecture. *ACM SIGCOMM Comput. Commun. Rev.* **2014**, *44*, 347–348. [CrossRef]
15. Fraire, J.A.; Madoery, P.G.; Finochietto, J.M. Traffic-aware contact plan design for disruption-tolerant space sensor networks. *Ad Hoc Netw.* **2016**, *47*, 41–52. [CrossRef]
16. Wang, P.; Li, H.; Zhang, S.; Lin, X.; Liu, J.; Wang, E. A Novel Joint Scheduling Scheme of Earth Observation and Transmission in Satellite Networks. In Proceedings of the 2020 International Conference on Computing, Networking and Communications (ICNC), Big Island, HI, USA, 17–20 February 2020; pp. 774–779.
17. Li, T.; Zhou, H.; Luo, H.; Quan, W.; Yu, S. Modeling software defined satellite networks using queueing theory. In Proceedings of the 2017 IEEE International Conference on Communications (ICC), Paris, Italy, 21–25 May 2017; pp. 1–6.
18. Li, T.; Zhou, H.; Luo, H.; Xu, Q.; Ye, Y. Using SDN and NFV to implement satellite communication networks. In Proceedings of the 2016 International Conference on Networking and Network Applications (NaNA), Hokkaido, Japan, 23–25 July 2016; pp. 131–134.
19. Zhou, D.; Sheng, M.; Wang, X.; Xu, C.; Liu, R.; Li, J. Mission aware contact plan design in resource-limited small satellite networks. *IEEE Trans. Commun.* **2017**, *65*, 2451–2466. [CrossRef]
20. Wang, Y.; Sheng, M.; Zhuang, W.; Zhang, S.; Zhang, N.; Li, J. Joint scheduling of observation and transmission in earth observation satellite networks. In Proceedings of the GLOBECOM 2017–2017 IEEE Global Communications Conference, Singapore, 4–8 December 2017; pp. 1–6.

21. Wang, Y.; Sheng, M.; Li, J.; Wang, X.; Liu, R.; Zhou, D. Dynamic contact plan design in broadband satellite networks with varying contact capacity. *IEEE Commun. Lett.* **2016**, *20*, 2410–2413. [CrossRef]
22. Zhou, D.; Sheng, M.; Li, B.; Li, J.; Han, Z. Distributionally robust planning for data delivery in distributed satellite cluster network. *IEEE Trans. Wirel. Commun.* **2019**, *18*, 3642–3657. [CrossRef]
23. He, L.; Li, J.; Sheng, M.; Liu, R.; Guo, K.; Zhou, D. Dynamic scheduling of hybrid tasks with time windows in data relay satellite networks. *IEEE Trans. Veh. Technol.* **2019**, *68*, 4989–5004. [CrossRef]
24. Ahmed, T.; Dubois, E.; Dupé, J.B.; Ferrús, R.; Gélard, P.; Kuhn, N. Software-defined satellite cloud RAN. *Int. J. Satell. Commun. Netw.* **2018**, *36*, 108–133. [CrossRef]
25. Bhoyar, D.; Kadam, M.; Sarode, P. Review of Software Defined Integrated Satellite-Terrestrial Network. In *ICCCE 2019*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 333–340.
26. Wang, Y.; Sheng, M.; Zhuang, W.; Zhang, S.; Zhang, N.; Liu, R.; Li, J. Multi-resource coordinate scheduling for earth observation in space information networks. *IEEE J. Sel. Areas Commun.* **2018**, *36*, 268–279. [CrossRef]
27. Du, J.; Jiang, C.; Wang, J.; Yu, S.; Ren, Y. Stability analysis and resource allocation for space-based multi-access systems. In Proceedings of the 2015 IEEE Global Communications Conference (GLOBECOM), San Diego, CA, USA, 6–10 December 2015; pp. 1–6.
28. Sheng, M.; Zhou, D.; Liu, R.; Wang, Y.; Li, J. Resource mobility in space information networks: Opportunities, challenges, and approaches. *IEEE Netw.* **2018**, *33*, 128–135. [CrossRef]
29. Liang, Q.; Fan, Y.; Yan, X.; Yan, Y. An algorithm based on differential evolution for satellite data transmission scheduling. *Int. J. Comput. Sci. Eng.* **2019**, *18*, 279–285. [CrossRef]
30. Wei, Z.; Zhao, B. A Space Information Service Forwarding Mechnism Based on Software Defined Network. *J. Internet Serv. Inf. Secur.* **2017**, *7*, 48–60.
31. Feng, B.; Zhou, H.; Zhang, H.; Li, G.; Li, H.; Yu, S.; Chao, H.C. HetNet: A flexible architecture for heterogeneous satellite-terrestrial networks. *IEEE Netw.* **2017**, *31*, 86–92. [CrossRef]
32. Du, J.; Jiang, C.; Guo, Q.; Guizani, M.; Ren, Y. Cooperative earth observation through complex space information networks. *IEEE Wirel. Commun.* **2016**, *23*, 136–144. [CrossRef]
33. Li, T.; Zhou, H.; Luo, H.; You, I.; Xu, Q. SAT-FLOW: Multi-strategy flow table management for software defined satellite networks. *IEEE Access* **2017**, *5*, 14952–14965. [CrossRef]
34. Li, T.; Zhou, H.; Luo, H.; Quan, W.; Xu, Q.; Li, G.; Li, G. Timeout strategy-based mobility management for software defined satellite networks. In Proceedings of the 2017 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Virtual, 1–4 May 2017; pp. 319–324.
35. Zhou, D.; Sheng, M.; Liu, R.; Wang, Y.; Li, J. Channel-aware mission scheduling in broadband data relay satellite networks. *IEEE J. Sel. Areas Commun.* **2018**, *36*, 1052–1064. [CrossRef]
36. Zhou, D.; Sheng, M.; Li, J.; Xu, C.; Liu, R.; Wang, Y. Toward high throughput contact plan design in resource-limited small satellite networks. In Proceedings of the 2016 IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC), Valencia, Spain, 4–8 September 2016; pp. 1–6.
37. Xu, S.; Wang, X.W.; Huang, M. Software-defined next-generation satellite networks: Architecture, challenges, and solutions. *IEEE Access* **2018**, *6*, 4027–4041. [CrossRef]
38. Bertaux, L.; Medjiah, S.; Berthou, P.; Abdellatif, S.; Hakiri, A.; Gelard, P.; Planchou, F.; Bruyere, M. Software defined networking and virtualization for broadband satellite networks. *IEEE Commun. Mag.* **2015**, *53*, 54–60. [CrossRef]
39. Fan, H.; Yang, Z.; Zhang, X.; Wu, S.; Long, J.; Liu, L. A novel multi-satellite and multi-task scheduling method based on task network graph aggregation. *Expert Syst. Appl.* **2022**, *205*, 117565. [CrossRef]
40. Tan, S.; Jin, F.; Xie, J.; Dun, C. A QoE Driven Cross-Domain Management Architecture for Space-Air-Ground Integrated Network. *Wirel. Commun. Mob. Comput.* **2022**, *2022*, 1–14. [CrossRef]
41. Bi, Y.; Han, G.; Xu, S.; Wang, X.; Lin, C.; Yu, Z.; Sun, P. Software defined space-terrestrial integrated networks: Architecture, challenges, and solutions. *IEEE Netw.* **2019**, *33*, 22–28. [CrossRef]
42. Fan, H.; Yang, Z.; Wu, S.; Zhang, X.; Long, J.; Liu, L. An Efficient Satellite Resource Cooperative Scheduling Method on Spatial Information Networks. *Mathematics* **2021**, *9*, 3293. [CrossRef]
43. Fan, H.; Long, J.; Liu, L.; Yang, Z. Dynamic Digital Twin and Online Scheduling for Contact Window Resources in Satellite Network. *IEEE Trans. Ind. Informatics* **2022** ; pp.1-10. [CrossRef]
44. San Martin, L.S.; Yang, J.; Liu, Y. Hybrid NSGA III/dual simplex approach to generation and transmission maintenance scheduling. *Int. J. Electr. Power Energy Syst.* **2022**, *135*, 107498. [CrossRef]
45. Chen, N.; Li, M.; Wang, M.; Su, Z.; Li, J.; Shen, X.S. A dynamic pricing based scheduling scheme for electric vehicles as mobile energy storages. In Proceedings of the ICC 2021-IEEE International Conference on Communications, Dublin, Ireland, 14–23 June 2021; pp. 1–6.
46. Rashidi, H. Simulation and Evaluation of Network Simplex Algorithm and its Extensions for Vehicle Scheduling Problems in Ports. *Int. J. Marit. Technol.* **2019**, *11*, 1–12. [CrossRef]

47. Zhang, Z.; Wu, W.; Yuan, J.; Du, D.Z. Breach-free sleep-wakeup scheduling for barrier coverage with heterogeneous wireless sensors. *IEEE/ACM Trans. Netw.* **2018**, *26*, 2404–2413. [CrossRef]
48. Hossein-Nejad, Z.; Agahi, H.; Mahmoodzadeh, A. Image matching based on the adaptive redundant keypoint elimination method in the SIFT algorithm. *Pattern Anal. Appl.* **2021**, *24*, 669–683. [CrossRef]