Chinese Journal of Electronics
vol. 33, no. 1, pp. 217–230, January 2024
https://doi.org/10.23919/cje.2022.00.114

RESEARCH ARTICLE

A Task Scheduling Algorithm Based on Clustering Pre-processing in Space-Based Information Network

Yufei WANG, Jun LIU, Shengnan ZHANG, Sai XU, and Jingyi WANG

School of Computer Science and Engineering, Northeastern University, Shenyang 110169, China

Corresponding author: Jun LIU, Email: liujun@cse.neu.edu.cn
Manuscript Received May 3, 2022; Accepted May 21, 2023

Abstract — With the diversification of space-based information network task requirements and the dramatic increase in demand, the efficient scheduling of various tasks in space-based information network becomes a new challenge. To address the problems of a limited number of resources and resource heterogeneity in the space-based information network, we propose a bilateral pre-processing model for tasks and resources in the scheduling pre-processing stage. We use an improved fuzzy clustering method to cluster tasks and resources and design coding rules and matching methods to match similar categories to improve the clustering effect. We propose a space-based information network task scheduling strategy based on an ant colony simulated annealing algorithm for the problems of high latency of space-based information network communication and high resource dynamics. The strategy can efficiently complete the task and resource matching and improve the task scheduling performance. The experimental results show that our proposed task scheduling strategy has less task execution time and higher resource utilization than other algorithms under the same experimental conditions. It has significantly improved scheduling performance.
Keywords — Space-based information network, Task scheduling, Resource clustering, Ant colony optimization, Simulated annealing.

Citation — Yufei WANG, Jun LIU, Shengnan ZHANG, *et al.*, "A Task Scheduling Algorithm Based on Clustering Pre-processing in Space-Based Information Network," *Chinese Journal of Electronics*, vol. 33, no. 1, pp. 217–230, 2024. doi: 10.23919/cje.2022.00.114.

I. Introduction

With the rapid development of space science and technology, space-based information networks are gradually becoming an essential infrastructure for the development of communication services [1]–[3]. The coverage of a space-based information network is relatively wide, which can realize the information collection and transmission between various geographic areas that are difficult to cover by terrestrial signals [4], [5].

With the emergence of low-cost constellation systems with large numbers of satellites such as StarLink, the space-based information network is faced with the problem that a task can be scheduled to multiple satellites. It is difficult for the space-based information network to achieve reasonable scheduling of tasks in multiple satellites. With the increase in the number of satellites, the types of tasks that can be served by the space-based information network have become diverse. Space-based information networks suffer from a limited number of on-board resources, heterogeneous on-board resources, high communication latency, and difficulty in upgrading on-board equipment [6]–[9]. Designing a task scheduling strategy applicable to space-based information networks is essential to address the above issues.

The existing research on resource scheduling of the space-based information network takes task demand as the input condition to schedule resources. To match resources that meet the requirements of task computing and storage, most of them are designed based on heuristic search algorithms. Li *et al.* [10] used the tabu search algorithm to solve the multi-satellite multi-task problem. The tabu search algorithm has weak global development ability and is easy to find the local optimal solution. Lin *et al.* [11] use a genetic algorithm to solve the multi-task

Associate Editor: Prof. Shangguang WANG, Beijing University of Posts and Telecommunications.

single satellite resource scheduling problem. He *et al.* [12] introduced ant colony optimization into the problem of matching satellite communication resources and tasks, which improved the comprehensive benefits of task scheduling. Most of the above studies analyze the whole satellite as a resource node and mission scheduling unit, lacking tasks and resource pre-processing. In contrast, the satellite as a scheduling unit can hardly meet the gradually growing mission demand. With the increasing number of tasks and satellite resources, the scheduling algorithm efficiency will decline.

To overcome the above shortcomings, a scheduling framework for large scale tasks and satellites is required. The scheduling framework should first simplify the scheduling size of resources and tasks. Secondly, it can make scheduling decisions quickly. Heuristic algorithms such as the ant colony optimization algorithm have good performance in path selection. However, there are different problems in the application of resource scheduling. In this paper, we propose a resource and task pre-processing model to reduce the scheduling scale of resources and tasks. To improve the problems in the resource scheduling process of the ant colony optimization algorithm. We also improved the ant colony algorithm for the resource characteristics of the space-based information network. To improve the problem that the ant colony optimization algorithm is easy to fall into the local optimum, we combine the simulated annealing algorithm to quickly jump out of the local optimum. It combines the advanced idea of a heuristic algorithm and improves the efficiency of resource utilization. The main contributions of this paper are as follows:

1) To reduce the task scheduling scale, we propose a pre-processing clustering algorithm to reduce the completion time of task scheduling according to task and resource attribute values. We use an improved fuzzy clustering algorithm to pre-process tasks and resources for clustering. We design coding rules and matching methods to reduce the search range for task assignment resources.

2) To improve the efficiency of resource utilization, we design the task scheduling strategy based on the ant colony simulated annealing algorithm according to the characteristics of space-based information network (SIN). We improve the scheduling performance by constructing a scheduling model and analyzing the scheduling objectives and designing factors.

3) We completed the simulation and results analysis. The results show that our proposed task scheduling strategy has better results in terms of task completion time and load balancing.

The remainder of this paper is organized as follows. Section II outlines the related works in this field. Section III introduces the system model and scheduling object. Section IV discusses the details of task scheduling pre-processing. Section V provides details about the proposed task scheduling algorithm. The performance of the system is simulated in Section VI. Finally, Section VII summarizes this paper.

II. Related Works

The research methodology of SIN task scheduling is divided into two aspects. The first aspect is the study of the architecture of SIN and the application of different techniques to improve the way SIN is managed. The second aspect is the design of scheduling algorithms for different situations, and researchers have applied different algorithms to design scheduling algorithms to improve the efficiency of task scheduling.

First, we discuss the researches on the architecture of SIN. In recent years, researchers have focused on integrated satellite networks formed by multi-star link interconnections, mainly applying SDN technology to simulate dynamic programmable reconfiguration of inter-star routing, network topology, and device resources as a way to improve integrated performance services [13]–[18]. Wang *et al.* proposed a resource management strategy for SIN to design scheduling and collaborative management based on dynamic space resource virtualization for multiple dimensions of resources in SIN [19]. Qu *et al.* proposed an architecture and network model for a time-space uninterrupted SIN and a SIN network model based on a hierarchical autonomous system [20]. Zhang *et al.* used cascaded fuzzy neural networks to design a new path selection algorithm by analyzing satellite payload resources, which can perform more efficient agile routing settings [21]. The resource mobility utilization strategy proposed by Sheng *et al.* effectively utilizes the practical resources that significantly improves the network performance [22]. Cheng *et al.* discussed the application of blockchain in SIN security for problems such as the vulnerability of nodes of space information networks to various cyber and physical attacks [23]. Zhang *et al.* designed a three-layer Walker constellation, a topology construction algorithm based on available time inter-satellite links to solve problems caused by satellite network dynamics [24]. Overall SIN architecture research has focused on virtualization, cloud computing, SDN, and layered architecture design. While these technologies are novel, research approaches rarely propose solutions for large-scale task and resource situations.

Secondly, the researchers mainly studied the design of the scheduling algorithm. With the gradual increase of user services the task requirements of users have shown a diversified trend and a certain increase in the number of missions. Researchers have focused on meeting the scheduling between large-scale tasks and resources and satellite architecture design [25]–[27]. More and more researchers are working on multi-satellite collaborative mission scheduling, and intelligent optimization algorithms are more effective in large-scale satellite mission scheduling. Habet *et al.* optimized the results by using a consistent percolation sampling method mainly in selecting the search space during resource allocation [28]. Sun *et al.* proposed an improved genetic algorithm to improve agile satellite scheduling [29]. Yuan *et al.* designed an enhanced genetic algorithm that sets high-quality initial solutions, resulting in a more efficient solution to the satellite mission scheduling scheme planning problem [30]. Li *et al.* used fuzzy neural networks and ant colony algorithms to solve the task scheduling problem [31]. Li *et al.* applied an improved genetic algorithm to the problem of scheduling satellite resources and task problems [32]. Huang *et al.* designed a scheduling strategy with an intelligent optimization algorithm for the tasks scheduling problem of multiple ground stations on multiple stars [33]. Sun *et al.* proposed a task scheduling mechanism for multiple users and tasks by studying space tasks and resources [34]. He *et al.* proposed a network swarm task scheduling algorithm based on double-threshold load balancing control to improve network process and memory management [35]. Meng *et al.* introduced the A3C algorithm in deep reinforcement learning to model and simulate the resource allocation process [36]. Wan *et al.* developed a multi-objective, multi-constrained network topology model for high network survivability and low link consumption, resulting in low link consumption and better convergence speed [37]. Chen *et al.* proposed a genetic algorithm based on population perturbation and elimination strategies focusing on the multi-satellite scheduling problem [38]. Scheduling algorithms mainly apply a heuristic search process, but fewer algorithms combine location factors and optimize for both resource usage and time consumption.

In our work, we have first considered the size of tasks and resources simplification problem. Secondly, the task scheduling algorithm is designed for the objectives such as resource usage and time consumption. Finally, to avoid the problem of local optimality of the algorithm, we combined different heuristic improvement ideas to optimize the scheduling algorithm.

III. System Model

In this section, we present the details of the resources and tasks considered in the system model, as well as the optimization goal of the system model. First, we introduce the resource virtualization system description. Secondly, we define the system description for task requests. Finally, the system utility goal is established in terms of both consumption time and resource usage.

1. Scheduling scenario

Our scenario is a scheduling model based on the cloud computing framework. The scheduling model is shown in Figure 1. The hardware resource layer is the resource entity of task scheduling, including high-orbit satellites, medium and low-orbit satellites, and ground cloud centers. The virtual resource layer is obtained based on the hardware resource layer. We cluster resources and schedule tasks according to virtual resources. The virtual resource set $\mathcal{R} = \{\boldsymbol{R}_1, \boldsymbol{R}_2, \ldots, \boldsymbol{R}_n\}$ and user request task set $\mathcal{T} = \{\boldsymbol{T}_1, \boldsymbol{T}_2, \ldots, \boldsymbol{T}_m\}$ in the system. The resource $\boldsymbol{R}_j = \{r_c, r_m, r_b\}$ contains three different resource capabilities computing, storage, and communication capabilities. Because of the dynamic characteristics of satellite networks, we assume that the topology of satellite networks is fixed in a small time slot, which is widely used for satellite network modeling [39]. We represent the space-based information network in the form of an aggregation graph $\boldsymbol{G} = (\mathcal{R}, E, [t_s, t_e])$. $\mathcal{R}$ is the resource node, E denotes that the resources are connected, and the connection time of each edge of $[t_s, t_e]$. The edge E_{ij} denotes the connection of the task from $\boldsymbol{R}_i$ to $\boldsymbol{R}_j$. When a task is published, the connection time C_{ij} between resource nodes $\boldsymbol{R}_i$ and $\boldsymbol{R}_j$ can be calculated according to E_{ij}. Finally, we can calculate all $\boldsymbol{C}$ according to all edges E in the network, and divide them into h non-overlapping time slots by time. The topology state of the space-based information network can be divided into time slot sequences $\boldsymbol{F} = \{[b_1, f_1], [b_2, f_2], \ldots, [b_h, f_h]\}$. In each slot $[b_i, f_i]$, we will schedule tasks to resources according to the scheduling algorithm. The next time slot starts to reschedule.

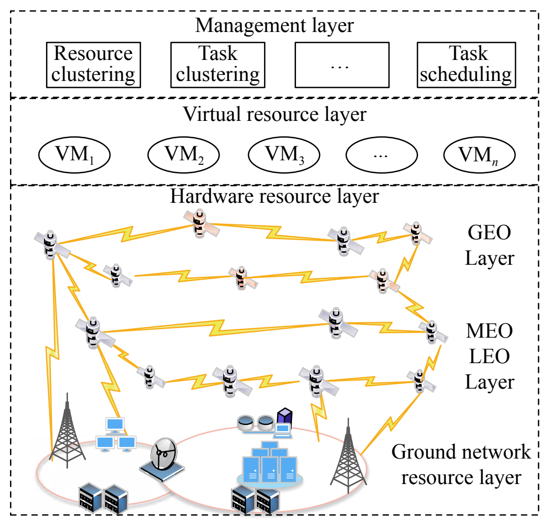


Figure 1 SIN cloud computing scheduling model.

2. Task model

Task requirements mainly include remote sensing, communication, and navigation tasks. At the same time, with the development of satellite network, taking StarLink commercial projects as an example, satellite networks began to access the Internet. The demands for satellite network tasks are more diverse. The main resource required by the remote sensing task is the storage resource and the main communication resource required by the communication task. Therefore, a single virtual resource form can better complete resource scheduling. When each time slot starts scheduling, the system contains the task set $\mathcal{T} = \{\boldsymbol{T}_1, \boldsymbol{T}_2, \ldots, \boldsymbol{T}_m\}$. We define each task as $\boldsymbol{T}_i = \{t_c, t_m, t_b\}$ according to the different requirements and resource forms of the task. Where t_c denotes the computing capability required for the task, t_m denotes the storage capacity required by the task, t_b denotes the communication capability required by the task.

3. Scheduling objective

As the number of satellites and tasks in the space-based information network increases, task scheduling time gradually increases while resource utilization efficiency decreases. Here, the task completion time cTime and the resource usage R^c of the task are considered as optimization objectives. The utility of the system can be defined as follows:

$$S = \alpha \sum_{\boldsymbol{T}_i \in \mathcal{T}} \mathrm{cTime}_i + \beta \sum_{\boldsymbol{T}_i \in \mathcal{T}, \boldsymbol{R}_j \in \mathcal{R}} \boldsymbol{R}_{ij}^c \tag{1}$$

where cTime_i denotes the time when the task T_i scheduling is completed. $\boldsymbol{R}_{ij}^c$ denotes the resource usage of the resource $\boldsymbol{R}_j$ used to complete the task $\boldsymbol{T}_i$. α and β are the weights of task scheduling completion time and resource usage, respectively.

The optimization objective of this paper minimizes the system utility, that is,

$$\begin{aligned} &\text{minimize } \boldsymbol{S} \\ &\text{s.t. } t_c^j \le r_c^j,\ \forall \boldsymbol{T}_i \to \boldsymbol{R}_j \\ &\quad\ \ t_m^j \le r_m^j,\ \forall \boldsymbol{T}_i \to \boldsymbol{R}_j \\ &\quad\ \ t_b^j \le r_b^j,\ \forall \boldsymbol{T}_i \to \boldsymbol{R}_j \end{aligned} \tag{2}$$

The above problem is a mixed nonlinear integer programming. It is usually an NP-hard problem. In this paper, we will propose a heuristic algorithm to solve the above problem.

IV. Space-Based Information Network Task Scheduling Pre-processing

The resource task bilateral clustering (RTBC) algorithm is proposed in the scheduling pre-processing stage, using an improved fuzzy clustering algorithm to cluster tasks and resources separately to obtain a certain number of task sets and resource sets. After the clustering is completed, the tasks and resources are matched to sets of the same type or similar types according to the similar class matching method.

1. Clustering model

1) Resource capacity vector definition

A resource has different kinds of capabilities. Therefore, the resource can be expressed as $\boldsymbol{R}_j = \{r_c, r_m, r_b\}$. r_c denotes the computing capacity of the resource, calculated by the number of instructions executed per second. r_m denotes the storage capacity of the resource, calculated by the amount of storage of the resource. r_b denotes the communication capacity of the resource, calculated by the bandwidth size of the resource.

The attribute demand vector is $\boldsymbol{V} = \{v_0, v_1, v_2\}$. v_0, v_1 and v_2 correspond to the demand vectors for computation, storage, and communication, respectively. $v_s \in \{-1, 0, 1\}$, $s = 1, 2, 3$, and $|v_s| = 1$ denotes interest in the first characteristic attribute indicator. The positive and negative of v_s denotes the direction. If $v_s>0$, the smaller the attribute indicator, the better. In contrust, if $v_s<0$, the larger the indicator is, the better. $v_s = 0$ denotes no interest in the s feature attribute indicator. The eigenvector can be obtained as

$$\boldsymbol{N}(\boldsymbol{R}_{k_c}) = (x_{k0}, x_{k1}, x_{k2}) \times \begin{pmatrix} |v_0| \\ |v_1| \\ |v_2| \end{pmatrix} \tag{3}$$

where x_{ks} denotes the s-th feature attribute indicator of the k-th resource $\boldsymbol{R}_k$. Where x_{k0} denotes the characteristic attribute metric of the k-th resource computing capability. x_{k1} denotes the characteristic attribute metric of the k-th resource storage capacity. x_{k2} denotes the characteristic attribute metric of the k-th resource transfer capability. $|v_j|$ denotes the absolute value of its corresponding demand vector. The resource feature attributes that are of interest can be left and those that are not can be eliminated by judging the $|v_s|$. Then the number of resource features is denoted as $|\boldsymbol{N}| = m = \{v_s | v_s \neq 0\}$.

2) Building data matrix

Each resource $\boldsymbol{R}_k$ in the $\mathcal{R} = \{\boldsymbol{R}_1, \boldsymbol{R}_2, \ldots, \boldsymbol{R}_n\}$ set of space-based information network resources has its own different resource properties, and each resource is represented by m attribute parameters for its resource properties, and the $n \times m$ raw data matrix $(x_{is})_{n\times m}$ of space-based information network resources can be obtained.

3) Data normalization

The attribute capability scales and orders of magnitude are different for each resource in the space-based information network resources. The attribute capability values of each resource are standardized to within the range of $[0,1]$ before the data are processed. According to the description of sensitivity in resource allocation by Wang *et al.* [40], the data normalization can be obtained by

$$x''_{ks} = \frac{x'_{ks} - \min\limits_{1\le k\le n}\{x'_{ks}\}}{\max\limits_{1\le k\le n}\{x'_{ks}\} - \min\limits_{1\le k\le n}\{x'_{ks}\}},\ (s = 1, 2, \ldots, m) \tag{4}$$

4) Building fuzzy similarity matrix

Different columns in the original data matrix are different resource attributes, and to classify resources with similar properties into one category, the exponential similarity coefficient method can be used to determine the degree of similarity between two resources, i.e., the similarity $P(\boldsymbol{R}_i, \boldsymbol{R}_j)$ between two different resources $\boldsymbol{R}_i$ and $\boldsymbol{R}_j$ is defined by

$$P(\boldsymbol{R}_i, \boldsymbol{R}_j) = p_{ij} = \frac{1}{m}\sum_{k=1}^{m} \mathrm{e}^{-\frac{3}{4}\cdot\frac{(x''_{ik} - x''_{jk})^2}{s''^2_k}} \in [0,1] \tag{5}$$

where $s''^2_k = \frac{1}{n}\sum_{k=1}^{n}(x''_{kj} - \overline{x''_j})^2$, $\overline{x''_j} = \frac{1}{n}\sum_1^n x''_{kj}$, $1\le i, j\le n$.

5) Cluster division

We use the optimized transfer closure method for clustering the resources, calculate the equivalence relation array $\boldsymbol{P}^*$, and complete the clustering by $\boldsymbol{P}^*$. The key to clustering is to make resource $\boldsymbol{R}_i$ and resource $\boldsymbol{R}_j$ belong to the same class at the level of division threshold α, and to specify the degree of equivalence between resource $\boldsymbol{R}_i$ and resource $\boldsymbol{R}_j$ is not less than α. The specific process of taking the optimal passing closure method is to set $\max = p_{ij}$ to compare each p_{jk} $(1 \le k \le n, k \neq i, j)$ with max in terms of numerical size, and assign the maximum value to max after each round of comparison. After optimizing the transfer closure method to solve each p^*_{ij}.

6) Setting division threshold α

As shown in the following equations (6) and (7), $\boldsymbol{P}^*_\alpha = p^{*(\alpha)}_{ij}$ is the intercept matrix of $\boldsymbol{P}^*$, and the resources $\boldsymbol{R}_i$ and $\boldsymbol{R}_j$ satisfying $p^{*(\alpha)}_{ij} = 1$ are grouped together to form an equivalence class, i.e., both are in the same logical subgroup.

$$p^{*(\alpha)}_{ij} = \begin{cases} 1, & p^*_{ij} \ge \alpha \\ 0, & p^*_{ij} < \alpha \end{cases} \tag{6}$$

$$[\boldsymbol{R}_i]_\alpha = \{\boldsymbol{R}_j | p^{*(\alpha)}_{ij} = 1\} \tag{7}$$

By setting different thresholds $\alpha(0 \le \alpha \le 1)$, it will be shown as a dynamic clustering graph depending on the clustering results. Refer to the analysis of the impact of different clustering radii on clustering results in the work [41]. We can get different α values that can effectively partition the clustering results. The smaller the value of α represents the lower the similarity between categories, and vice versa, the higher the similarity, thus forming different division intervals $\{[1, \alpha_x), [\alpha_x, \alpha_y), \dots, [\alpha_z, o]\}$.

The component $\overline{\boldsymbol{R}_k}$ of $\mathcal{R} = \{\boldsymbol{R}_1, \boldsymbol{R}_2, \dots, \boldsymbol{R}_n\}$ needs to be a value between $[0, 1]$, specified by

$$\overline{\boldsymbol{R}_k} = \frac{\boldsymbol{R}_k}{\sum_{i=1}^{n} \boldsymbol{R}_i} \tag{8}$$

where $1 \le k \le n$, $\overline{\boldsymbol{R}_k}$ is the attribute weight of the space-based information network resources. The minimum error tolerance vector $\boldsymbol{E} = (e_1, e_2, \dots, e_n)$ is given because there are differences in the attribute capabilities between resources due to the heterogeneity between resources. Where e_i is the minimum error tolerance of the resource of class i and $0 \le e_i \le 1$. The size of this value affects its error and can indicate the importance of the factor, the larger the e_i the more important, so the formula for calculating the division threshold α of the resource is obtained by

$$\alpha = \sum_{k=1}^{n} \boldsymbol{R}_k \times e_k \tag{9}$$

7) Clustering performance modelling

For an arbitrary resource $\boldsymbol{R}_k$, the resource characteristic attribute demand vector $\boldsymbol{V} = \{v_0, v_1, v_2\}$ and the resource demand weight $\boldsymbol{W} = \{w_0, w_1, w_2\}$ are used to represent the preference of $\boldsymbol{R}_k$ for different resource characteristic attributes.

Let η be the number of classifications after the clustering is completed, the i-th resource class in the resource classification is noted as RL_i, there are n_i resources in each resource class, and the comprehensive attribute capability of the i-th resource class is defined by

$$CP(\mathbf{RL}_i) = \frac{1}{n_i} \sum_{\boldsymbol{R}_k \in \mathbf{RL}_i} \sum_{s=1}^{m} v_s \cdot w_s \cdot x''_{ks} \tag{10}$$

where v_s denotes the direction of the demand vector of the attribute feature of the s-dimensional resource, and w_s denotes the demand weight of the attribute feature of the s-dimensional resource.

2. Similar category matching

After clustering the task set and resource set separately using the above improved fuzzy clustering method, the process of similar category matching is performed next. For example, ten original resources are clustered into five resource classes. One of the possible results of resource clustering is shown in Table 1 to represent each of the five different resource types.

Table 1 Resource clustering results

r_{a_0}	r_{a_1}	r_{a_2}	r_{a_3}	r_{a_4}
$\boldsymbol{R}_3, \boldsymbol{R}_6$	$\boldsymbol{R}_1, \boldsymbol{R}_4, \boldsymbol{R}_7$	$\boldsymbol{R}_2, \boldsymbol{R}_{10}$	$\boldsymbol{R}_9$	$\boldsymbol{R}_5, \boldsymbol{R}_8$

At the end of clustering, the features of the class center within the same class can approximately represent the features of other objects in the class, so the features of the class center are used to represent the data object features of the class.

We calculate the size of the attribute values of the cluster center's demand for computation, storage, and bandwidth, i.e., we judge the demandability of the current cluster center for these three dimensions. Suppose the value of the dimension attribute of the cluster center exceeds 60% of its value range. In that case, the demand for the current dimension is considered vital, and this attribute is set as a strong attribute, and the essential attribute is coded as "1" and vice versa, it is coded as "0". Tasks that are computation-intensive and I/O-intensive image or video processing are stronger in terms of CPU and bandwidth requirements but do not consume much memory when processing this type of task, and this type of task can be coded as "110", as shown in Table 2.

The Hemming distance is used as the difference degree measure between the task class and the resource class. For example, if the class code where t is located is "010" and the class code where n is located is "011", the difference measure between "010" and "011" has a dif-

Table 2 Coding example

Type	Computing	Storage	Communication	code
$\boldsymbol{R}_1$	2000	100	200	'010'
$\boldsymbol{R}_2$	4500	200	300	'110'

ferent measure of 1.

The specific similar category matching method is as follows: the nearest Hemming distance priority matching is used as the rule for similar matching between categories, taking the task class as the reference, the first matching is the resource class with a Hemming distance of 0 from the task class, then the resource class with a Hemming distance of 1 from the task class and so on. And finally, all task classes will find a suitable resource class. For example, assuming that tasks and resources are classified into five classes using a modified fuzzy clustering algorithm, the difference degree matrices between these 5 task class codes and five resource class codes and between task and resource classes are shown in Table 3.

Table 3 Difference matrix

Type	r_{a_0}("010")	r_{a_1}("110")	r_{a_2}("011")	r_{a_3}("000")	r_{a_4}("001")
t_{a_0}("110")	1	0	2	2	3
t_{a_1}("011")	1	2	0	2	2
t_{a_2}("101")	3	2	2	2	1
t_{a_3}("111")	2	1	1	3	2
t_{a_4}("010")	0	1	1	1	2

According to the above similar category matching method, firstly, the task class with the different degree of 0 and the resource class are matched first, i.e., t_{a_0} matches r_{a_1}, t_{a_1} matches r_{a_2}, t_{a_4} matches r_{a_0}, secondly, the task class with the different degree of 1 and the resource class are selected, then t_{a_2} matches r_{a_4}, and finally, the t_{a_3} task class with the different degree of 2 and the r_{a_4} resource class are selected and matched, and the results are shown in Table 4.

Table 4 Similar matching results between sets

1	2	3	4	5
$[t_{a_0}, r_{a_1}]$	$[t_{a_1}, r_{a_2}]$	$[t_{a_4}, r_{a_0}]$	$[t_{a_2}, r_{a_4}]$	$[t_{a_3}, r_{a_4}]$

The bilateral pre-processing model based on the improved fuzzy clustering algorithm is mainly based on categorizing the own demand for tasks and the attribute characteristics of resources, forming a certain number of task categories and resource categories, and prioritizing the matching of similar types of task categories and resource categories. The RTBC pseudo code is shown in Algorithm 1.

Algorithm 1 The implementation of the RTBC algorithm

Input: $\mathcal{T}$: Tasks' set; $\mathcal{R}$: Resources' set; num: Iteration number; a: Number of task clusters; b: Number of resource cluster.

Output:

c: matching pairs of task sets and resource sets.

```
Data normalization 𝒯, ℛ;
while Iteration number < num do
  Clustering operation for 𝒯,ℛ;
end while
Get a set of task clusters T_list, b set of resources R_list;
Calculating the difference degree value dis;
min ← 0;
while R_list ≠ null do
  if dis_ij = min then
    Matching task set T_i to resource set R_j;
    T_list delete task set T_i;
    R_list delete resource set R_j;
  end if
  min ← min + 1;
end while
```

According to the above algorithm description, we can obtain the time complexity of the algorithm. The algorithm time complexity is determined through Big O Notation. The RTBC algorithm need two iterations to obtain the final result. The first iteration is to cluster $\mathcal{T}$ and $\mathcal{R}$, with an iteration number of m and time complexity is $O(m)$. The similarity matching of resource and task clustering results is judged circularly according to the $\boldsymbol{R}_{\text{list}}$ quantity n of the resource list. The time complexity of this step is $O(n)$. Hence, the total time complexity of the RTBC clustering algorithm is $O(m+n)$.

V. Task Scheduling Algorithm

After the tasks and resources are pre-processed by clustering, a suitable scheduling algorithm is needed to schedule the tasks to the appropriate resources for execution. We design an improved ant colony optimization simulated annealing (IACOSA) task scheduling algorithm based on the characteristics of the space-based information network with the optimization goal of reducing task scheduling completion time and improving resource utilization.

1. Scheduling constraints

1) Scheduling constraints

The resources of SIN are heterogeneous. For different resources, the capabilities of each attribute of their resources are different, so the response time of the same task varies on different resources. The total response time of a single task T_i on the resource R_j is expressed by

$$\mathrm{cTime}_{ij} = \mathrm{wTime}_{ij} + \mathrm{kTime}_{ij} \tag{11}$$

where wTime_{ij} denotes the time that the task needs to wait before executing on the resource and is expressed in equation (12). tTime_{ij} is a process that requires a transfer before a task can be assigned to a resource for execution, and is calculated by (13). a denotes the number of tasks that have been assigned on resource j. t_{i_b} denotes that bandwidth is required before the task i can be executed. r_{j_b} denotes the size of the bandwidth in the resource j attribute. kTime_{ij} refers to the time that a task is expected to execute after it is deployed on the resource set and is calculated by (14).

$$\text{wTime}_{ij} = \begin{cases} \text{tTime}_{ij}, \\ \quad \text{unassigned tasks on resource nodes} \\ \sum_{x=1}^{a} \text{kTime}_{xj} + \text{tTime}_{ij}, \\ \quad \text{otherwise } 1 \le x \le a \le m \end{cases} \tag{12}$$

$$\text{tTime}_{ij} = \frac{t_{i\_b}}{r_{j\_b}} \tag{13}$$

$$\text{kTime}_{ij} = \frac{T_e}{R_p} \tag{14}$$

The task execution time kTime_{ij} is calculated as the ratio of the resource profile T_e required for task i execution and the capacity R_p that resource j can provide.

After clustering, tasks of the same type, depending on their demand areas, will be concurrently executed mapped to matched sets of similar resources. If we want to minimize the execution completion time of all tasks in SIN, we have to ensure that the overall completion time of all tasks refers to The maximum value of the task completion time on a single resource. That is, the maximum value in cTime_{ij}, the task execution completion time of a certain allocation scheme K can be expressed by

$$\text{cTime}(K) = \max(\text{cTime}),\ i \in [1, m],\ j \in [1, n] \tag{15}$$

As the demand of each task is different, assigning all tasks to the resource with the weakest attribute capability to execute, the task completion time is the largest, denoted by $\text{cTime}_{\max}$. On the contrary, assigning all tasks to the resource with the strongest attribute capability, the task completion time is the smallest, denoted by $\text{cTime}_{\min}$.

2) Location constraints

The location of satellite nodes in SIN is dynamically changing, and each resource class is able to serve only a certain range in a certain time period due to the different locations of resources in the task scheduling process. The distance situation between the demand area of task i and the service range of resource j can be expressed in terms of G_{ij}, which is calculated by

$$G_{ij} = \sqrt{(\text{lon}_i - \text{lon}_j)^2 + (\text{lat}_i - \text{lat}_j)^2 + (h_i - h_j)^2} \tag{16}$$

where lon, lat, and h denote longitude, latitude, and altitude respectively. A smaller value of G_{ij} denotes that the gap between the two is smaller, and the closer the demand area of task i and the service range of resource j are, the smaller the scheduling completion time of the task will be.

3) Credibility factor

Trustworthiness refers to the ability to provide services for a resource within a certain time frame, and this trustworthiness varies with time, and the trustworthiness takes values in $[0, 1]$. Trustworthiness means that at a given moment T_1, the resource $\boldsymbol{R}_j$ is derived based on the experience of processing task type t_x in the record of historical behavior, and each resource needs to maintain a table of trustworthiness relations, which is calculated by

$$\gamma(T_1 - T_x, t_k) = \mathrm{e}^{-(T_1 - T_x)/n(t_k)} \tag{17}$$

where, t_k is the decay rate of confidence. T_1 denotes the current time. T_x denotes the last execution time of the task type t_x on the resource. A larger value of T_x means a more drastic degree of credibility decay. The credibility factor is calculated by

$$D(R_j, t_x, T_1) = \left| \frac{n_x - s_x}{n_x + s_x} \right| \cdot \gamma(T_1 - T_x, t_x) \tag{18}$$

where n_x denotes the number of times resource $\boldsymbol{R}_j$ has successfully completed task type t_x. s_x denotes the number of failures of resource $\boldsymbol{R}_j$ for task type t_x. The trustworthiness of the tasks in the queue to be assigned to the resource changes from time to time and from task to task, and the trustworthiness needs to be updated in real time. When a resource successfully performs a task, it needs to increase the trustworthiness of the resource for that task type, and conversely decrease the trustworthiness if it fails.

Based on the above idea, the resource uses the following equation to update its credibility:

$$D(\boldsymbol{R}_j, t_x, T_1) = \begin{cases} D(\boldsymbol{R}_j, t_x, T_2) + \lambda_1 \mathrm{e}^{-1/v}, \\ \quad (0 < \lambda_1 < 1) \wedge (N = 1) \\ D(\boldsymbol{R}_j, t_x, T_2) - \lambda_2 \mathrm{e}^{-1/v}, \\ \quad (0 < \lambda_2 < 1) \wedge (N = 0) \end{cases} \tag{19}$$

$$D(\boldsymbol{R}_j, t_x, T_2) = \begin{cases} 1, & D(\boldsymbol{R}_j, t_x, T_1) > 1 \\ 0, & D(\boldsymbol{R}_j, t_x, T_1) < 0 \end{cases} \tag{20}$$

where $D(\boldsymbol{R}_j, t_x, T_1)$ is the confidence level obtained from the evaluation of resource $\boldsymbol{R}_j$ after the completion of task type t_x at moment T_1. $D(\boldsymbol{R}_j, t_x, T_2)$ denotes the credibility before the update, as shown in equation (20). λ_1 and λ_2 denote the update factor. v denotes the number of resource $\boldsymbol{R}_j$ completions for the task type t_x. N denotes the flag of success or failure of this execution task. If $N = 1$, it means resource $\boldsymbol{R}_j$ successfully executes task type t_x, otherwise it fails.

2. Task scheduling algorithm based on ant colony simulated annealing algorithm

The matching process of tasks and resources is a mixed nonlinear integer programming with complex variables. According to similar work [42], this is usually an NP-hard problem. The heuristic algorithm can find an approximate solution to this kind of problem. Because the matching process from task to resource is similar to path selection, we choose the ant colony optimization algorithm as the basis of task scheduling. But for SIN, we need to make targeted improvements. In the process of improvement, the simulated annealing algorithm is combined to improve the overall performance of the algorithm.

The resources of SIN are structured through network connectivity. The pheromone of the ant colony algorithm needs to consider the performance of task execution. The performance of different resources varies greatly. The pheromone in the initial state needs to be set according to the specific application scenario. In the design of heuristic information for the algorithm, some factors that can affect the effect of task execution need to be considered. Therefore, through the corresponding improvement of the basic algorithm, the improved algorithm can be suitable for solving the task scheduling problem of the space-based information network and achieve a better scheduling effect.

Ant colony optimization (ACO) algorithm has high flexibility and robustness, but the algorithm has the defects of slow convergence and is easy to occur into local optimum. Simulated annealing (SA) has the ad-vantage of better local search capability and fast convergence, which can make up for the defects of the ACO algorithm. We propose an improved ant colony simulated annealing algorithm (IACOSA) with the characteris-tics of a space-based information network. It can thus better serve the space-based information network and achieve the expected goal of task scheduling. The specific algorithm design is as follows.

1) Initialization method design

The original ant colony optimization algorithm sets the initial pheromone as the same value. To accelerate the convergence of the algorithm, we improve the initialization of pheromones by applying task and resource locations and resource capabilities. The initial value of the pheromone of the path search is determined jointly with the attribute capabilities of the resource and the location information of the resource. The specific calculation is shown as

$$\tau_j(0) = R_c + \frac{1}{dr_{ij}} \tag{21}$$

where, R_c denotes the attribute capability of the resource. dr_{ij} denotes the location relationship of the resource. In the beginning of the algorithm, since the pheromones on each path are different, according to the attribute capability of the resources and the location information, the ants can make purposeful path selection and avoid the problem of strong randomness at the beginning of the algorithm.

2) The way ants select resources

Ant colony optimization algorithm is an algorithm for routing based on path probability [42]. We designed pheromone and heuristic information in an ant colony optimization algorithm for the optimization goal of matching tasks to resources in the space-based information network. Through this probability, we can find the matching path from the optimal task to the resource that meets the optimization goal. Ants need to perform a path search based on state migration rules. When selecting a resource, ant k takes into account the location factor and the trustworthiness of the resource, and then calculates the transfer probability by using pheromones and heuristic information to select the next selected resource. The migration probability of task T_i, i.e., ant k, between resources R_j at the moment of t is expressed by $p_{ij}^k(t)$ as

$$p_{ij}^k(t) = \begin{cases} \dfrac{[\tau_{ij}(t)]^\alpha [\eta_{ij}(t)]^\beta}{\sum_{s=1}^{m} [\tau_{is}(t)]^\alpha [\eta_{is}(t)]^\beta}, & s, j \in \text{allowed}_k \\ 0, & \text{otherwise} \end{cases} \tag{22}$$

Here $\tau_{ij}(t)$ is the pheromone concentration of the task $\boldsymbol{T}_i$ scheduled to the resource $\boldsymbol{R}_j$ for execution. $\eta_{ij}(t) = \frac{1}{G_{ij}} + D(\boldsymbol{R}_j, t_i, T_1)$ represents heuristic information. Both G_{ij} and $D(\boldsymbol{R}_j, t_i, T_1)$ are heuristic factors, representing the location distance situation and the degree of credibility, respectively. The coefficients α, β denote the weight indexes in the control pheromone and state transfer probability, respectively. A smaller G_{ij} represents the proximity of the task demand area to the range served by the resource. A higher value of $D(\boldsymbol{R}_j, t_i, T_1)$ denotes a higher level of confidence in the resource. The task performs the selection of resources in allowed_k, and when the ant has finished searching for a resource, it gets rid of it from allowed_k.

3) Pheromone update

The pheromone concentration will influence the pheromone concentration in the path selection. The pheromones on the path are changed and updated with each task deployment. $\tau_{ij}(t)$ is the pheromone value on the path from task $\boldsymbol{T}_i$ to resource $\boldsymbol{R}_j$ at the moment of t, and the pheromone update at the moment of $t+1$ is calculated by

$$\tau_{ij}(t+1) = (1-\rho) \cdot \tau_{ij}(t) + \Delta\tau_{ij}(t),\ \rho \in [0, 1) \tag{23}$$

where, the parameter ρ refers to the pheromone volatility factor, which represents the volatility degree of pheromone concentration in the process of constant pheromone updating, and the larger parameter ρ means the faster volatility.

The time constraint function is improved to the update rule of the pheromone, and the following equation is used to evaluate the degree of time spent for the selected task scheduling scheme "K":

$$\mathrm{sTime}(K) = \frac{\mathrm{cTime}(K) - \mathrm{cTime}(K)_{\min}}{\mathrm{cTime}(K)_{\max} - \mathrm{cTime}(K)_{\min}} \tag{24}$$

The time constraint function is proportional to the time required to complete the task execution, and the smaller the value, the shorter the execution time of the group of tasks. The design of the pheromone update increment $\Delta\tau_{ij}(t)$ needs to consider the task completion execution time during the task scheduling process of SIN, so the calculation of the pheromone increment $\Delta\tau_{ij}(t)$ can be expressed as

$$\Delta\tau_{ij}(t) = \frac{Q_1}{\mathrm{sTime}(K)} \tag{25}$$

where Q_1 is a constant, the pheromone increment in (25) is the increment used in the local pheromone update.

The local optimal solution of ACO is obtained as the initial solution of SA algorithm, and the pheromone update of the optimal solution derived from SA algorithm is performed to obtain the global pheromone increment, which is calculated by

$$\Delta\tau_{ij}(t) = \frac{Q_2}{\min[\mathrm{sTime}(K)]} \tag{26}$$

where Q_2 is a constant, and the global pheromone increment is to find the best solution for pheromone update among all the scheduling solutions.

4) Metropolis guidelines

If the resources allocated for these two tasks are different, the resources are swapped based on the local optimal solution. If the completion time of the task is reduced after the swap, then this new solution is accepted. Otherwise, it is judged whether to receive this new solution according to the Metropolis criterion of simulated annealing. According to the following equations (27) and (28), the probability p of using the new solution is calculated. If the value of p is smaller than the random value generated at the current temperature T, then this new solution will not be accepted, and vice versa.

$$\Delta\mathrm{cTime\_T} = \mathrm{cTime\_T} - \mathrm{cTime}_{\min}\mathrm{\_T} \tag{27}$$

$$p_{ij}^{k}(t) = \begin{cases} \mathrm{e}^{-\left(\frac{\mathrm{cTime\_\Delta T}}{T}\right)}, & \mathrm{cTime\_\Delta T} > 0 \\ 1, & \text{others} \end{cases} \tag{28}$$

where cTime_T denotes the sum of the time that all tasks run after swapping resources at the current temperature T. $\mathrm{cTime}_{\min}\mathrm{\_T}$ denotes the minimum time used to execute all resources to complete all tasks at the current temperature T using the ant colony algorithm. p denotes the probability that a new solution is acceptable when the conditions of $\Delta\mathrm{cTime\_T} > 0$ are satisfied.

In the sampling process of SA algorithm, if the current local optimal solution is disturbed at the same temperature for M consecutive times, if this local optimal solution always does not change in any way, then it can be considered to be in full compliance with the criterion of sampling stability. The termination condition of the algorithm can be derived by analyzing the annealing process according to the simulated annealing process, i.e., comparing the magnitude of the current temperature T and $T_{\min}$, when $T(t+1) < T_{\min}$, then the algorithm is terminated.

Based on the above analysis and design, the IACOSA algorithm is described in detail by pseudo code, as shown in Algorithm 2.

Algorithm 2 IACOSA algorithm

Input:

$\mathcal{T} = \{\boldsymbol{T}_1, \boldsymbol{T}_2, \ldots, \boldsymbol{T}_m\}$, $\mathcal{R} = \{\boldsymbol{R}_1, \boldsymbol{R}_2, \ldots, \boldsymbol{R}_n\}$.

Output:

Optimal scheduling solution for tasks and resources.

```
1: Set α, β, ρ, λ, Q_1, Q_2, T_0, T_min;
2: Initializing pheromone τ_j(0) = R_c + 1/dr_ij
3: Calculating the path transfer probability p_ij(t);
4: for T_i ∈ 𝒯 do
5:   for R_j ∈ ℛ do
6:     Calculating the local optimal solution according to cTime_min;
7:     Updating local pheromone based on τ_ij(t+1);
8:     A new solution to the construction of the Metropolis criterion in SA;
9:     Calculating the probability of accepting the new solution p;
10:    Updating Global pheromone;
11:    if T(t+1) > T_min then
12:    goto step 5
13:    else
14:    return the optimal solution;
15:    end if
16:  end for
17: end for
```

According to the design process of Algorithm 2, we can get the time complexity of the algorithm. The number of tasks in the IACOSA algorithm is n and the number of resources is m. The time complexity of task-to-resource scheduling process is $O(nm)$. The path selection of the ant colony optimization algorithm is done by accumulating pheromones by ants. The number of ants is k. The time complexity is $O(knm)$. The path selection needs several iterations to complete, and the number of iterations is l. From the above analysis, we finally get the algorithm time complexity as $O(lknm)$.

Refer to Chen *et al.* [43] for a detailed analysis of the algorithm response time. We perform a specific calculation of the overhead time of the algorithm. When the number of ants $m = 10$, the number of satellite tasks takes the average size $n = 200$ and the number of resources $m=40$. It can be calculated that the task scheduling time $t=18.3$ s. The motion of satellites is periodic, and generally the orbital altitude of low orbiting satellites is around 500–2000 km. The orbital period is $T=2\pi\sqrt{r^3/\mu}$, where $\mu=398600.441\ \mathrm{km}^3/\mathrm{s}^2$, r is the orbital height. The operating period can be calculated from 5663.7 s to 76173 s. At the same time, the satellite flight process covers the ground, and when the satellite side field of view is 30°, the coverage area radius can be calculated as 288.68 km. The flight speed of the satellite at 500 km orbital altitude is 7.6 km/s. We can roughly calculate the time taken by the satellite to fly over the ground coverage area to be t = 75.96 s. First of all, we can compare that our scheduling time is much smaller than the satellite cycle time. Also, the satellite scheduling time is smaller than the time of the satellite flying over a region. Therefore, our proposed algorithm can be applied to satellite networks with high dynamic changes.

VI. Simulation and Analysis

1. Simulation environment and scenario establishment

We established a SIN scenario consisting of 6 GEOs + 14 MEOs + 10 LEOs + 5 ground stations by Satellite Tool Kit (STK), as shown in Figure 2. We can get the resource locations of different periods by STK and slice the satellite's operation time into different time slices to get different resource topology and location information.

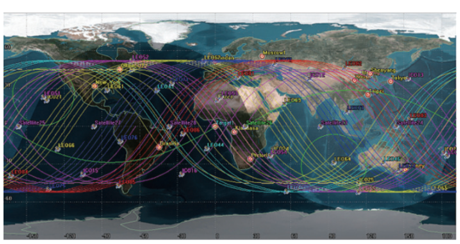


Figure 2 Space-based information network scenario.

We perform the simulation verification of the scheduling algorithm by CloudSim cloud simulation software. In addition to the resource location information and topology information, we need to configure the simulation's task and resource capability parameters. The resources of SIN are set through VM instances in CloudSim, and the tasks of SIN can be set through Task instances in CloudSim. The CloudSim simulation environment resources include VM, Host, and datacenter classes. We use VM as the resource clustering and scheduling unit. The satellites are set as Host, GEO server, MEO server, and Ground server are set as Datacenter, and in addition to the inherent resource parameters of ram, bw and storage in CloudSim, we add the time slot t, latitude, longitude, and high parameters. Based on these parameters, we can determine resource connectivity, resource location, etc. The same three parameters are included in the task settings. The requested location is for example the various ground targets in Figure 2. Based on the resource and task parameters we can perform scheduling planning. The location and moment parameters of the resources we obtain according to STK. Other parameters are shown in Table 5.

Table 5 Cloudsim platform parameter configuration

Type	Parameters	Value
Resource sets	Number	40
	CPU (MIPS)	[100, 10500]
	Storage (MB)	[10000, 1050000]
	Communication (MB)	[100, 10000]
	Manager type	Time-Sharing
Task sets	Number	[10,600]
	Length (MIPS)	[100, 5000]
	Storage (MB)	[10000, 460000]
	Communication (MB)	[100, 10000]

The parameter settings of the ACO algorithm for the experimental simulation are shown in Table 6, where IACOSA sets the parameters $T_0=1000$, $T_{\min}=10$, $a=0.995$, $M=50$ respectively on the basis of ACO.

Table 6 ACO parameters

Parameters	Value	Parameters	Value
α	1	m	37
β	2	Q_1, Q_2	100
ρ	0.4	$I_{\max}$	100

2. Analysis of simulation performance

Since the algorithm proposed in this paper is a heuristic algorithm, the optimal matching results obtained from the algorithm optimization vary somewhat with the algorithm's solution process. We repeated the simulations 10 times with the same variables for the accuracy of the results, and the average of the statistical results is shown in the resulting graph. Also, for the comprehensiveness of the experimental results, we show the standard deviation SD between different simulation results and the mean value in the form of error bars in the experimental results.

1) Simulation performance analysis of different improvement ideas

We analyze the impact of different improvement ideas on the simulation results. The improvement ideas include the improved ant colony optimization algorithm (IACO), the improved ant colony optimization simulated annealing algorithm (IACOSA) and the improved ant colony optimization simulated annealing algorithm based on RTBC clustering (RTBC-IACOSA). As shown in Figure 3(a) and (b), the performance of different improvement ideas in task scheduling completion time and resource usage is shown respectively. It can be seen from the results that RTBC-IACOSA performs best. From the paper, we can know that IACO only improved the ant colony algorithm to adapt to the characteristics of the space-based information network. However, IACO is prone to fall into local optimum and perform the worst. After the combination of the simulated annealing algorithm and ant colony algorithm, IACOSA can quickly jump out of the local optimum, and the result is better than IACO. After adding RTBC clustering, tasks and resources can be matched faster, so RTBC-IACOSA performs best.

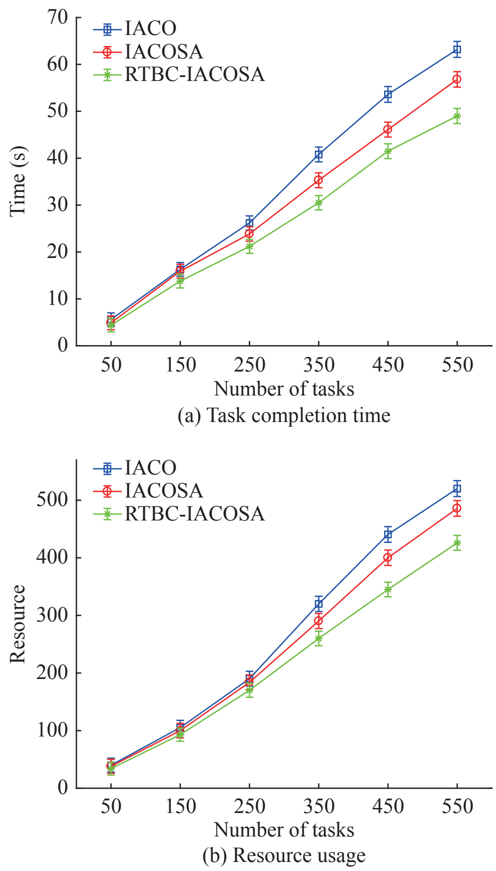


Figure 3 Comparison of different improvement ideas.

2) Scheduling performance analysis

We simulate and analyze the proposed RTBC-IACOSA, a joint RTBC and IACOSA-based task scheduling strategy for space-based information networks. The RTBC-IACOSA algorithm is based on the IACOSA algorithm with the addition of RTBC clustering. Therefore, the ACO algorithm [42] is selected among the heuristic algorithms for comparison. In contrast, the Min-Min algorithm [44] that comes with the CloudSim platform is selected among the traditional task scheduling algorithms for comparison with it. At the same time, we compare the existing satellite network resource scheduling algorithm PSOW [45]. The RTBC-IACOSA algorithm is compared with these algorithms in terms of the four performance evaluation indexes of scheduling completion time, resource usage, load balance degree, and failed services. The effectiveness and advantages of our RTBC-IACOSA algorithm in task scheduling are verified.

ACO simulates the path selection process of ants in nature when they are looking for food. Ants leave pheromones on their search paths, and pheromones can guide ants to choose paths. The path selection process of ants is similar to that of a task to resource matching. This paper is mainly about the improvement of this algorithm. We take ACO as the benchmark algorithm for comparison. Min-Min is a more traditional and classical scheduling algorithm. The main scheduling idea of the algorithm is to allocate and process tasks in the fastest time. Assign tasks to the resources with the shortest processing time. PSOW (particle swarm optimization weight) is a population based optimization algorithm that simulates the collective behavior of birds. These groups will cooperate to find food, and the group can improve the efficiency of searching for food by learning about the experience of each member. The learning process can be controlled by the weight factor to achieve rapid convergence.

As shown in Figure 4, the task scheduling completion time are shown for a different number of tasks using three different algorithms. It can be seen that as the number of tasks increases the task scheduling completion time increases and the scheduling completion time used in the scheduling scheme obtained using the RTBC-IACOSA algorithm is reduced to a great extent compared to the ACO, PSOW, and the Min-Min algorithm. By improving the heuristic factor of the ACO algorithm in the RTBC-IACOSA algorithm, the tasks can quickly find the resources close to their locations and reduce the time consumption due to the location relationship. By introducing the confidence factor, the tasks can be reasonably matched with the resources faster.

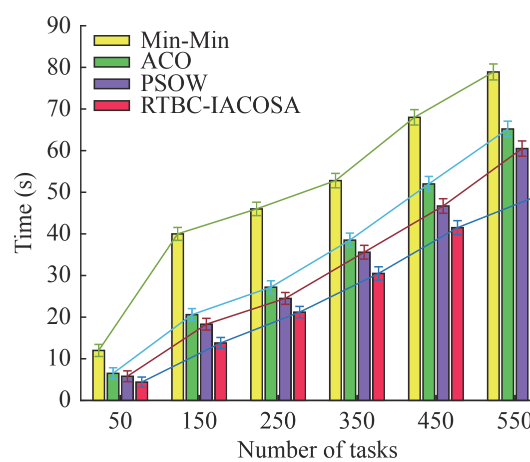


Figure 4 The task scheduling completion time with the number of tasks.

As shown in Figure 5, the variation of the sum of resource usage for each task corresponding to the use of three different algorithms at a different number of tasks is shown. The RTBC-IACOSA algorithm still has an advantage over the ACO, PSOW, and Min-Min algorithms.

This is because the RTBC-IACOSA algorithm clusters resources and tasks using a bilateral pre-processing model in the scheduling pre-processing stage. Tasks are matched with resources in a similar type of resource set, resulting in lower usage of resources.

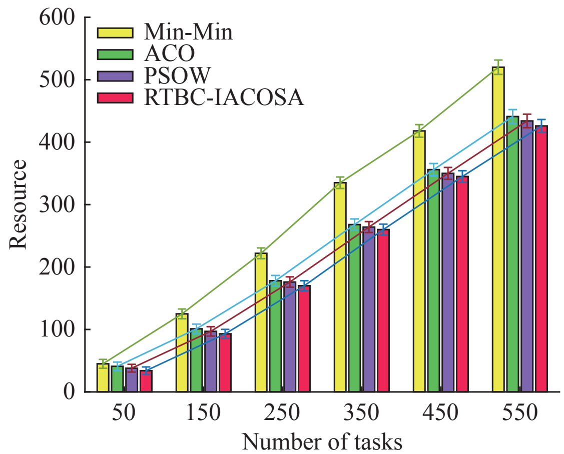


Figure 5 Changes in resource usage with the number of tasks.

As in Figure 6 (a), the relative standard deviation is used to measure the degree of load balancing of the algorithms. The changes in the relative standard deviation of the RTBC-IACOSA scheduling algorithm, ACO, PSOW, and Min-Min algorithms are compared separately as the number of tasks increases. The difference is not significant because RTBC-IACOSA, as a kind of swarm intelligence algorithm, has a certain global search ability and distributes tasks relatively evenly. Compared with the ACO, PSOW, and Min-Min algorithms, the scheduling scheme generated by RTBC-IACOSA is always in the best state of load balancing and more stable, with improved performance, and the algorithm is optimal.

As Figure 6 (b) compares the number of failed services of different algorithms, the number of failed service tasks can effectively measure the algorithm's reliability. From the results, the number of failed service requests for the four algorithms Min-Min, ACO, PSOW and RTBC-IACOSA gradually increases when the number of tasks gradually increases. However, compared with the other three algorithms, the RTBC-IACOSA algorithm always has the lowest number of failed requests due to introducing a confidence factor in the ant state transfer probability when designing the RTBC-IACOSA algorithm, which makes the algorithm have better reliability.

VII. Conclusions

This paper proposes a clustering pre-processing based task scheduling strategy for SIN. A bilateral clustering pre-processing model is proposed for the problem of long initial resource and task search time for task scheduling. To improve the matching effect of tasks and resources and enhance resource utilization, we propose a task scheduling algorithm with improved ant colony simulated annealing, which first improves the pheromone

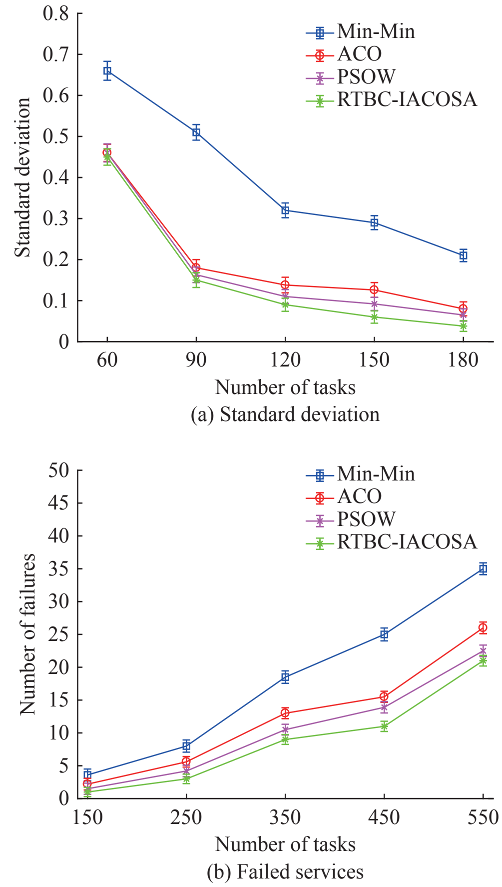


Figure 6 Change curve with the number of tasks.

and heuristic factor of the traditional ant colony algorithm and introduces the simulated annealing algorithm to avoid the local optimum situation. After simulation verification, our proposed task scheduling strategy can improve task scheduling performance, effectively reduce task scheduling time and increase resource utilization.

Acknowledgement

This work was supported by the National Natural Science Foundation of China (Grant Nos. 62071134 and 61671141) and the Fundamental Research Funds for the Central Universities (Grant Nos. N2116015 and N2116 020).

References

[1] L. H. Chi, C. W. Lin, W. H. Lin, *et al.*, "Research on development of space-ground integration information network," in *Proceedings of 2020 International Conference on Urban Engineering and Management Science (ICUEMS)*, Taiyuan, China, pp. 29–32, 2020.

[2] M. Tropea, F. De Rango, and A. F. Santamaria, "Design of a two-stage scheduling scheme for dvb-S2/S2X satellite architecture," *IEEE Transactions on Broadcasting*, vol. 67, no. 2, pp. 424–437, 2021.

[3] A. Y. Alhilal, T. Braud, and P. Hui, "A roadmap toward a unified space communication architecture," *IEEE Access*, vol. 9, pp. 99633–99650, 2021.

[4] H. J. Pan, H. P. Yao, T. L. Mai, *et al*, "Scalable traffic control using programmable data planes in a space information network," *IEEE Network*, vol. 35, no. 4, pp. 35–41, 2021.

[5] H. T. Yang, W. Liu, H. Y. Li, *et al*, "Maximum flow routing strategy for space information network with service function constraints," *IEEE Transactions on Wireless Communications*, vol. 21, no. 5, pp. 2909–2923, 2022.

[6] C. Q. Dai, C. Li, S. Fu, *et al*, "Dynamic scheduling for emergency tasks in space data relay network," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 1, pp. 795–807, 2021.

[7] K. P. Xue, T. De Cola, D. S. L. Wei, *et al*, "Guest editorial: Space information networks: Technological challenges, design issues, and solutions," *IEEE Network*, vol. 35, no. 4, pp. 16–18, 2021.

[8] S. J. Ji, M. Sheng, D. Zhou, *et al*, "Flexible and distributed mobility management for integrated terrestrial-satellite networks: Challenges, architectures, and approaches," *IEEE Network*, vol. 35, no. 4, pp. 73–81, 2021.

[9] C. Q. Dai, S. P. Li, J. S. Wu, *et al*, "Distributed user association with grouping in satellite–terrestrial integrated networks," *IEEE Internet of Things Journal*, vol. 9, no. 12, pp. 10244–10256, 2022.

[10] H. J. Li, Y. Lu, F. H. Dong, *et al*, "Communications satellite multi-satellite multi-task scheduling," *Procedia Engineering*, vol. 29, pp. 3143–3148, 2012.

[11] Y. S. Lin, H. L. Jiang, Y. L. Dong, *et al*, "Research of dynamic scheduling method for communication satellite resources based on genetic algorithm," *Radio Engineering*, vol. 47, no. 6, pp. 20–23, 2017.

[12] Y. He, H. Y. Zhang, and Z. Ren, "Research on GEO satellite communication resources-task analysis and modeling match," *Aerospace Control*, vol. 32, no. 6, pp. 44–49,56, 2014.

[13] J. Zhang and X. S. Yu, "Control and routing of satellite communication port in integrated space-ground networks," *Radio Communications Technology*, vol. 43, no. 2, pp. 1–5, 2017.

[14] Z. G. Liu, J. Zhu, J. M. Zhang, *et al*, "Routing algorithm design of satellite network architecture based on SDN and ICN," *International Journal of Satellite Communications and Networking*, vol. 38, no. 1, pp. 1–15, 2020.

[15] T. X. Li, H. C. Zhou, H. B. Luo, *et al*, "Service: A software defined framework for integrated space-terrestrial satellite communication," *IEEE Transactions on Mobile Computing*, vol. 17, no. 3, pp. 703–716, 2018.

[16] X. L. Meng, L. D. Wu, J. Jiao, *et al.*, "Research on resource allocation method of the sin based on SDN," in *Proceedings of 2019 IEEE International Geoscience and Remote Sensing Symposium*, Yokohama, Japan, pp. 10071–10074, 2019.

[17] P. Qin, H. J. Liu, X. N. Zhao, *et al.*, "Software-defined space-based integration network architecture," in *International Conference in Communications, Signal Processing, and Systems*, Dalian, China, pp. 449–458, 2020.

[18] Z. Q. Wang, B. J. Cui, S. Yao, *et al.*, "An SDN-based dynamic security architecture for space information networks," in *4th International Conference on Space Information Network*, Wuzhen, China, pp. 99–111, 2019.

[19] R. Wang, X. D. Han, N. Xu, *et al.*, "Resource scheduling and cooperative management of space information networks," in *Third International Conference on Space Information Network*, Changchun, China, pp. 146–151, 2019.

[20] Z. C. Qu, G. X. Zhang, T. Hong, *et al*, "Architecture and network model of time-space uninterrupted space information network," IEEE Access, vol. 7, pp. 27677–27688, 2019.

[21] Y. Y. Zhang, Y. C. Shi, Y. H. Pang, *et al.*, "Research on path selection for space network based on fuzzy neural network," in *Proceedings of 2019 11th International Conference on Wireless Communications and Signal Processing (WCSP)*, Xi'an, China, pp. 1–6, 2019.

[22] M. Sheng, D. Zhou, R. Z. Liu, *et al*, "Resource mobility in space information networks: Opportunities, challenges, and approaches," *IEEE Network*, vol. 33, no. 1, pp. 128–135, 2019.

[23] S. C. Cheng, Y. Gao, X. Y. Li, *et al.*, "Blockchain application in space information network security," in *Third International Conference on Space Information Network*, Changchun, China, pp. 3–9, 2019.

[24] Y. Zhang, B. Wang, B. L. Guo, *et al*, "A research on integrated space-ground information network simulation platform based on SDN," *Computer Networks*, vol. 188, pp. 107821, 2021.

[25] X. N. Niu, H. Tang, and L. X. Wu, "Satellite scheduling of large areal tasks for rapid response to natural disaster using a multi-objective genetic algorithm," *International Journal of Disaster Risk Reduction*, vol. 28, pp. 813–825, 2018.

[26] J. F. Cordeau and G. Laporte, "Maximizing the value of an earth observation satellite orbit," *Journal of the Operational Research Society*, vol. 56, no. 8, pp. 962–968, 2005.

[27] J. Berger, N. Lo, and M. Barkaoui, "Quest–a new quadratic decision model for the multi-satellite scheduling problem," *Computers & Operations Research*, vol. 115, pp. 104822, 2020.

[28] D. Habet, M. Vasquez, and Y. Vimont, "Bounding the optimum for the problem of scheduling the photographs of an agile earth observing satellite," *Computational Optimization and Applications*, vol. 47, no. 2, pp. 307–333, 2010.

[29] K. Sun, Z. Y. Yang, P. Wang, *et al*, "Mission planning and action planning for agile earth-observing satellite with genetic algorithm," *Journal of Harbin Institute of Technology*, vol. 20, no. 5, pp. 51–56, 2013.

[30] Z. Yuan, Y. W. Chen, and R. J. He, "Agile earth observing satellites mission planning using genetic algorithm based on high quality initial solutions," in *Proceedings of 2014 IEEE Congress on Evolutionary Computation (CEC)*, Beijing, China, pp. 603–609, 2014.

[31] Y. Q. Li, R. X. Wang, and M. Q. Xu, "Rescheduling of observing spacecraft using fuzzy neural network and ant colony algorithm," *Chinese Journal of Aeronautics*, vol. 27, no. 3, pp. 678–687, 2014.

[32] Y. Q. Li, R. X. Wang, Y. Liu, *et al*, "Satellite range scheduling with the priority constraint: An improved genetic algorithm using a station ID encoding method," *Chinese Journal of Aeronautics*, vol. 28, no. 3, pp. 789–803, 2015.

[33] M. S. Huang, S. Y. Chen, Y. Zhu, *et al*, "Topology control for time-evolving and predictable delay-tolerant networks," *IEEE Transactions on Computers*, vol. 62, no. 11, pp. 2308–2321, 2013.

[34] F. Sun, L. Gui, and H. P. Chen, "Task flow based spatial information network resource scheduling," in *Second International Conference on Space Information Network*, Yinchuan, China, pp. 249–258, 2018.

[35] J. J. He, "Improvement of efficient scheduling algorithm for network massive resources," in *Proceedings of 2018 International Conference on Virtual Reality and Intelligent Systems (ICVRIS)*, Hunan, China, pp. 528–531, 2018.

[36] X. L. Meng, L. D. Wu, and S. B. Yu, "Research on resource allocation method of space information networks based on deep reinforcement learning," *Remote Sensing*, vol. 11, no. 4, pp. 448, 2019.

[37] S. M. Wan, P. Yang, M. Zhuo, *et al.*, "An improved space information networks topology algorithm," in *Proceedings of 2021 IEEE International Conference on Information Communication and Software Engineering (ICICSE)*, Chengdu, China, pp. 227–232, 2021.

[38] M. Chen, J. Wen, Y. J. Song, *et al*, "A population perturbation and elimination strategy based genetic algorithm for multi-satellite TT&C scheduling problem," *Swarm and Evolutionary Computation*, vol. 65, pp. 100912, 2021.

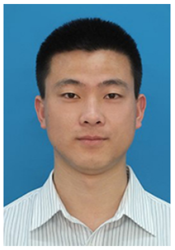
[39] Z. Y. Jia, M. Sheng, J. D. Li, *et al*, "Joint hap access and

LEO satellite backhaul in 6G: Matching game-based approaches," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 4, pp. 1147–1159, 2021.
[40] F. Wang, D. D. Jiang, S. Qi, *et al.* "A dynamic resource scheduling scheme in edge computing satellite networks," *Mobile Networks and Applications*, vol. 26, no. 2, pp. 597–608, 2021.
[41] M. Naghibzadeh, H. Taheri, and P. Neamatollahi, "Fuzzy-based clustering solution for hot spot problem in wireless sensor networks," in *Proceedings of 7'th International Symposium on Telecommunications (IST'2014)*, Tehran, Iran, pp. 729–734, 2014.
[42] Z. J. Zhang, F. N. Hu, and N. Zhang, "Ant colony algorithm for satellite control resource scheduling problem," *Applied Intelligence*, vol. 48, no. 10, pp. 3295–3305, 2018.
[43] H. K. Chen, X. Zhang, L. Wang, *et al*, "Resource-constrained self-organized optimization for near-real-time offloading satellite earth observation big data," *Knowledge-Based Systems*, vol. 253, pp. 109496, 2022.
[44] N. Sharma, S. Tyagi, and S. Atri, "A comparative analysis of min-min and max-min algorithms based on the makespan parameter," *International Journal of Advanced Research in Computer Science*, vol. 8, no. 3, pp. 1038–1041, 2017.
[45] S. Lu, F. Zheng, W. Si, *et al.*, "Computation resource allocation based on particle swarm optimization for LEO satellite networks," in *Proceedings of the 9th International Conference on Computer Engineering and Networks*, Changsha, China, pp. 955–962, 2021.

Yufei WANG received the B.S. degree in communication engineering from Yanshan University, Qinhuangdao, China, in 2016, and the M.S. degree in electronic and communication engineering from Northeastern University, Shenyang, China, in 2020. He is currently a Ph.D. candidate in School of Computer Science and Engineering, Northeastern University, Shenyang, China. His current research interests include resource management and space information networks.
(Email: yfwang@stumail.neu.edu.cn)

Jun LIU received the Ph.D. degree in communications and information systems from the Northeastern University. He is currently a Professor and Ph.D. Supervisor with Northeastern University. His research interests include space information networks, selforganizing networks, and information security.
(Email: liujun@cse.neu.edu.cn)

Shengnan ZHANG received the B.S. degree in electronic information engineering from Dalian Nationalities University, Dalian, China, in 2018, and the M.S. degree in electronic and communication engineering from Northeastern University, Shenyang, China, in 2021. Her research direction includes space-based information network and resource scheduling.
(Email: zsn112900@163.com)



Sai XU received the M.S. degree in computer software and theory from Northeastern University, Shenyang, China, in 2014. He is a Ph.D. candidate in computer technology with the School of Computer Science and Engineering, Northeastern University, Shenyang, China. His research interests include cloud computing and edge computing.
(Email: xu_s@neusoft.com)

Jingyi WANG received the B.S. degree in communication engineering from Northeastern University, Shenyang, China, in 2021. She is an M.S. candidate in information and communication engineering with the school of Northeastern University, Shenyang, China. Her research interests include resource scheduling and space information network.
(Email: wangjingyimmx@foxmail.com)