Advances in Astronautics Science and Technology
https://doi.org/10.1007/s42423-024-00165-z

ORIGINAL PAPER

A Conflict-Priority-Based Variable Neighborhood Tabu Search Method for Multi-satellite Scheduling

Xiaoyu Chen[1,2] · Qi Gao[1,2] · Shengren Peng[3] · Shihui Song[3] · Yufei Liu[3] · Guangming Dai[1,2] · Maocai Wang[1,2] · Changxing Zhang[1,2]

Received: 9 May 2024 / Revised: 11 June 2024 / Accepted: 5 July 2024

Abstract

With the increasing number and capabilities of orbital satellites and their onboard sensors, satellites scheduling plays an important role in growing demands for Earth observation tasks. A conflict-priority-based variable neighborhood tabu search method is proposed in this paper. By analyzing the flexibility of resources, the conflict-priority of tasks, and the visible time window features, the indicators of the impact, the conflict, the flexibility, and the bad impact are first established. On this basis, a first-come first-served greedy algorithm and a minimum bad impact greedy algorithm are designed for the generation of the initial solution. some rule-based heuristic strategies are also adopted for optimizing. Additionally, the algorithm incorporates three types of neighborhood structures including insertion, swapping and rearrangement, and deletion, using a tabu list to avoid local optima. Experimental results indicate the validity and efficiency of the proposed method. All solutions are within 10% of the optimal solution, with some even within 5%.The proposed method is effective for obtaining optimum solutions or solutions with a very good quality in various multi-satellite scheduling scenarios of realistic planning.

Keywords Multi-satellite cooperation · Mission scheduling · Variable neighborhood tabu search · Mission conflict degree · Resource flexibility

1 Introduction

Earth observation satellites primarily focus on observing designated ground targets according to mission requirements [1] and transmitting the data to ground receiving stations. In recent years, the demand for observations under various topographic and spectral conditions has rapidly increased [2, 3], leading to a pressing need for satellite resources. Satellite imaging scheduling involves selecting observation tasks to be executed within a given time frame and arranging satellite resources and timing to maximize user demand satisfaction.

The multi-satellite cooperative scheduling is a resource and execution time cooperated optimization problem, which has been proven to be NP-hard. It involves optimizing the allocation of satellite sensor resources and scheduling the acquisition, processing, and transmission of information while meeting the constraints of time windows according to user demands for satellite application tasks. Typically, the

✉ Maocai Wang
cugwmc@163.com

Xiaoyu Chen
xiaoyu.chen@cug.edu.cn

Qi Gao
cuggaoqi@gmail.com

Shengren Peng
785751053@qq.com

Shihui Song
sshh_1990@sina.com

Yufei Liu
liuyufeihit@sina.com

Guangming Dai
cugdgm@126.com

Changxing Zhang
zhangchx2023@163.com

[1] School of Computer, China University of Geosciences, Lumo Road, Wuhan 430074, People's Republic of China

[2] Hubei Key Laboratory of Intelligent Geo-Information Processing, China University of Geosciences, Lumo Road, Wuhan 430074, People's Republic of China

[3] Qian Xuesen Laboratory, China Academy of Space Technology, Youyi Road, Beijing 100094, People's Republic of China

Published online: 26 July 2024

objective is to complete as many tasks as possible within a certain time window or consider task profits to maximize total revenue, thereby minimizing losses. The satellite task scheduling problem in implementation needs to consider various factors such as limitations of satellite resources, satellite trajectories [4, 5], satellite attitudes [6], types of satellite sensors, task priorities, and time window restrictions. Among these, the accuracy and scientific nature of satellite trajectory planning are directly related to the success of satellite missions. The precision and real-time control of satellite attitude significantly affect the efficiency and effectiveness of satellite scheduling. Additionally, satellite resources, task priorities, and the visible time window directly impact the final task scheduling order. These factors can be contradictory, and trade-offs are necessary to achieve the optimal solution. To address these issues, researchers have proposed many effective algorithms such as greedy algorithms, dynamic programming algorithms, genetic algorithms, etc., which have shown certain effects in practical applications.

This paper designs a variable neighborhood tabu search algorithm for the problem of multi-satellite cooperative scheduling, aiming to enhance the capability and efficiency of task arrangement in such problems. The study focuses on three main aspects: (1) Selection of either all or part of the results from the greedy algorithm as the initial solutions for the tabu search algorithm, comparing the convergence speed and the quality of the final solutions under different initial conditions. (2) Utilization of a variable neighborhood tabu search algorithm that, during operations such as insertion and replacement of tasks, employs strategies prioritizing conflict and minimizing bad impact, explicitly evolving towards a more relaxed resource usage. (3) Compare the gap between the algorithm results and the ideal optimal solution to evaluate the algorithm performance.

The remainder of the paper is organized as follows. In Sect. 2, we review the related literature. Section 3 provides an overview of the multi-satellite cooperative scheduling problem. Section 4 designs indicators for the selection of tasks and their observation time windows, describes the calculation of these windows, including methods for union and complement as well as usage scenarios, and introduces four evaluation indicators for tasks or windows: impact, conflict, flexibility, and bad impact. Section 5 provides a detailed introduction to the proposed conflict-priority-based variable neighborhood tabu search algorithm and the methods for analyzing the results of the algorithm. Experiments and the analysis of results are reported in Sect. 6. Finally, Sect. 7 concludes the paper.

2 Related Works

To address the problem of multi-satellite cooperative task planning and considering operational constraints such as task execution time dependencies, Chen et al. constructed a mixed-integer linear programming model incorporating cutting-plane techniques [7], and proposed an improved differential evolution algorithm based on task priority and conflict resolution, which enhances the generation and efficiency of optimal solutions for large-scale task cooperative planning [8]. Furthermore, Chen et al. [9] conducted a thorough analysis of scene resources and the spatiotemporal characteristics of tasks within multi-satellite cooperative planning, developing a unified formal system architecture tailored for various complex observation challenges. Additionally, taking into account operational constraints including task execution time dependencies, Chen et al. [10] proposed a data-driven, two-stage multi-satellite data transmission cooperative planning method.

This paper opts for the tabu search algorithm to solve the multi-satellite cooperative scheduling issue. The tabu search algorithm was initially proposed by Glover [11] and was detailed as a combinatorial optimization algorithm [12]. The tabu search algorithm has been applied across various domains, demonstrating its wide applicability. Li et al. [13] employed the tabu search algorithm to solve the traveling salesman problem with precedence relationships. Cheng et al. [14] used a taboo search algorithm to solve a product-service scheduling problem that combines multiple traveling salesman problems and service matching for multiple service types. Chen et al. [15] combined the tabu search algorithm with the genetic algorithm to address the location problem of distribution centers. Kang et al. [16] used the tabu search algorithm to solve the NP-hard problem of circular packing. tabu search has also been applied in power distribution network planning [17–19]. Award et al. [20] used the tabu search algorithm to address timetabling issues. Saleh et al. [21] described the steps of the tabu search algorithm applied to the Global Positioning System surveying network. Rahdar et al. [22] proposed the application of a variable neighborhood tabu search algorithm to the bus stop selection problem.

In using tabu search algorithms to address satellite scheduling planning problems, Bianchessi et al. [23] discussed the scheduling of the multi-satellite constellation and established an integer programming model for the problem but used an inexact tabu search algorithm for solving it. Zuo et al. [24] combined the concept of variable neighborhood search with tabu search, but the initial solution used a general greedy solution targeting maximum profit, resulting in slower solving speed. Li et al. [25] further improved the solving efficiency of tabu search algorithms in satellite scheduling problems, but did not consider the rearrangement of replaced tasks during neighborhood searches.Yang et al. [26] used a fast insertion tabu search with conflict avoidance heuristics to solve the multi-satellite multi-mode cross-link scheduling problem, they used the conflict avoidance method to generate a high-quality initial solution and then optimized the initial solution using a forbidden search algorithm.

The multi-satellite cooperative scheduling, as an NP-hard problem with significant application value, is difficult to solve within an acceptable time using ordinary traditional algorithms. The development of intelligent optimization algorithms, such as genetic algorithms, particle swarm optimization, and ant colony algorithms, has propelled further optimization in solving problems of multi-satellite cooperative scheduling. Since the introduction of the tabu search algorithm, this type of intelligent optimization algorithm has been widely applied to classic and new applied problems. However, when using the tabu search algorithm to solve the multi-satellite scheduling problems, in order to make the algorithm have a better global optimality search capability and convergence, it should not only be considered to be limited to a fixed neighborhood and a fixed length of the tabu list, or the use of a variable neighborhood exploration in the incorporation of unguided stochasticity, the degree of neighborhood exploration is shallow, resulting in a generally slower convergence speed and miss the better solution.

3 The Multi-Satellite Cooperative Scheduling

3.1 Problem Description

The problem of multi-satellite cooperative scheduling focuses on the scheduling and observation of ground targets by satellite sensors. In this scenario, there are multiple ground targets with observation needs, each having attributes of minimum continuous observation time and associated profits. Multiple satellites equipped with various types of sensors orbit the Earth and can observe targets within a certain radius centered on the ground. The potential observation periods for ground targets by these sensors within a simulation period are known as visible time windows. These time windows may overlap with each other, and since sensors can only observe one ground target at any given moment, there is a possibility of mutual exclusivity among the tasks. The challenge for the algorithm is to maximize either the total value of profits or the maximum number of observable targets, all while avoiding conflicts among the tasks (Fig. 1).

3.2 Scheduling Model

3.2.1 Symbolic Description

Scheduling Simulation Periods. Tasks are planned within the simulation time interval $[S_{Beg}, S_{End}]$, where S_{Beg} is the earliest start time for sensor allocation and S_{End} is the latest end time for sensor allocation.

Task Set $M = \{M_1, M_2, ..., M_n\}$. Each task represents a ground target, with a minimum required observation duration. Each task needs to be continuously observed within the sensor's available time window, and each ground target corresponds to an execution profit.

Sensor Set $R = \{R_1, R_2, ..., R_n\}$, also known as the resource set. Continuous observation of two tasks on sensor R_j requires a minimum transition time, $tran_j$.

Visible Time Slice Set $TW = \{tw_{i,j}^1, tw_{i,j}^2, ..., tw_{i,j}^{N_{i,j}}\}$, where each visible time window $tw_{i,j}^k$ has an earliest start time $Beg_{i,j}^k$ and a latest end time $End_{i,j}^k$. This represents the collection of assignable time windows for all available sensors R_j for target M_i within a simulation periods.

The continuous allocable time segments for sensors are denoted as $RTW = \{rtw_1, rtw_2, ..., rtw_n\}$. By calculating the intersection of the visible time windows for each task on every sensor resource, one can obtain the collection of available time intervals $RTW_j = \{rtw_j^1, rtw_j^2, ..., rtw_j^{N_j}\}$ for the corresponding sensor within a period. For all $tw_{i,j}^k$, if $M_i \in M(R_j)$,then $tw_{i,j}^k \subset RTW_j$.

3.2.2 Optimization Variables

(1) For each observable ground target M_i and sensor R_j, the variable $x_{i,j}^k$ denotes the status of assigning target M_i to the kth visible time window of sensor R_j. If $x_{i,j}^k = 1$, it indicates that the task is allocated to the visible time window $tw_{i,j}^k$.

(2) After allocating an execution time window to the target M_i, the start time of execution is denoted as t_i, and the end time of execution is the start time plus the duration D_i of the task.

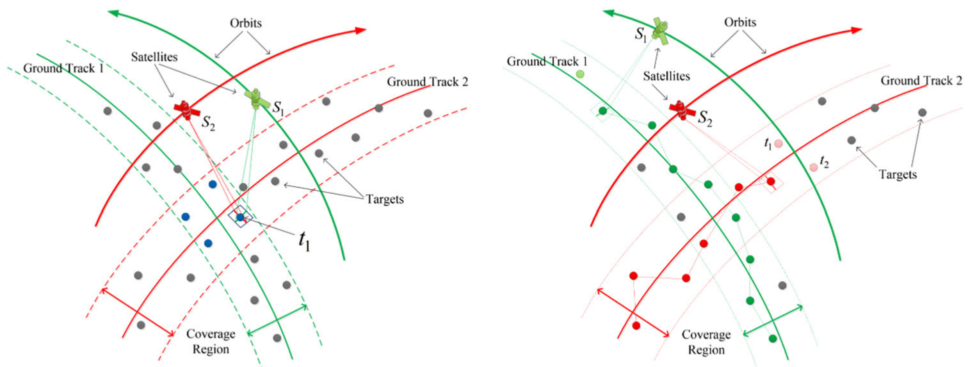
3.2.3 Optimization Objective

(1) Maximize the total number of tasks completed.

$$max \sum_{M_i \in M} \sum_{R_j \in R(M_i)} \sum_{k \in 1,2,...,N_{i,j}} x_{i,j}^k \tag{1}$$

(2) Maximize the total profits of task execution.

$$max \sum_{M_i \in M} \sum_{R_j \in R(M_i)} \sum_{k \in 1,2,...,N_{i,j}} W_i \cdot x_{i,j}^k \tag{2}$$



(a) Observation resource selection

(b) Resource contention conflict

Fig. 1 Processing of the multi-satellite scheduling [8]

3.2.4 Constraints Analysis

(1) Number of task execution. Each task, even if observable by multiple windows of multiple sensors throughout the entire simulation periods, is observed at most once. For each task $M_i \in M$, we have:

$$max \sum_{R_j \in R(M_i)} \sum_{k \in 1,2,\ldots,N_{i,j}} x_{i,j}^k \le 1 \tag{3}$$

(2) Feasibility of task execution start time. Every ground target $M_i \in M$ needs to be assigned within the sensor's earliest observable time and latest end time period.

$$S_{Beg} \le t_i \le S_{End} - D_i \tag{4}$$

(3) Observation window constraint. The execution time interval allocated for each ground target must meet the sensor's availability condition and the ground target's minimum imaging duration constraint. Within the simulation periods, if the ground target M_i is allocated sensor R_j and the corresponding visible time window $tw_{i,j}^k$, then the task's execution time interval must fall entirely within the assigned visible time window. For any $M_i \in M$, and any $R_j \in R(M_i)$, $k \in 1, 2, \ldots, N_{i,j}$, this constraint can be represented as:

$$Beg_{i,j}^k \le t_i \le End_{i,j}^k - D_i, \quad \text{if } x_{i,j}^k = 1. \tag{5}$$

(4) The minimum transition time between two consecutive tasks on the same sensor. During the process of observing ground targets, the side-swing angle and rotation angle of the onboard resources change. Thus, two tasks arranged to be executed sequentially on the same sensor must satisfy the minimum transition time between them.

Therefore, for any $R_j \in R$ and any two consecutive observation task sequences $M_i, M_{i'} \in M(R_j)$, considering the different execution order of the two tasks, the expression 6 is introduced to represent the temporal constraints before and after the execution of two tasks.

$$t_i - t_{i'} \ge (D_{i'} + \Delta_{i,i'}^j) \ \text{ or } \ t_{i'} - t_i \ge (D_i + \Delta_{i',i}^j) \tag{6}$$

Since each target may have multiple visible time windows on multiple sensors, the $Big - M$ (the value depends on the size of the scheduling simulation period.) can be adopted to eliminate the constraint conflicts that occur when multiple candidate time windows are considered for selection at the same time. The modified observation window constraint can be represented as Eq. (7).

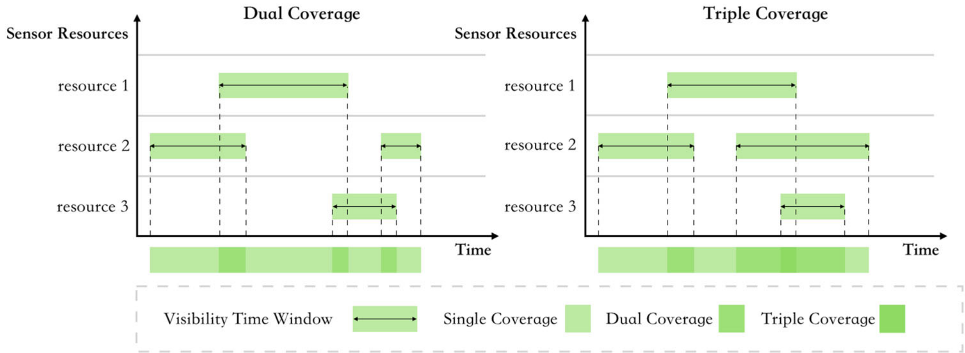
$$\begin{aligned} & t_i - Beg_{i,j}^k \cdot x_{i,j}^k \ge 0 \\ & t_i - (End_{i,j}^k - D_i) \cdot x_{i,j}^k - M \cdot (1 - x_{i,j}^k) \le 0. \end{aligned} \tag{7}$$

Additionally, for the constraint of the minimum transition time between two consecutive tasks on the same sensor, since only one condition needs to be met to eliminate the "or" condition in the expression, we introduce two binary variables $f_{i,i'}^j$ and $f_{i',i}^j$, with $f_{i,i'}^j + f_{i',i}^j = \sum_k x_{i,j}^k \cdot \sum_{k'} x_{i',j}^{k'}$, this constraint can be expressed as:

$$\begin{aligned} t_i - t_{i'} &\ge (D_{i'} + \Delta_{i,i'}^j) \cdot f_{i,i'}^j - (M - D_{i'}) \cdot (1 - f_{i,i'}^j) \\ t_{i'} - t_i &\ge (D_i + \Delta_{i',i}^j) \cdot f_{i',i}^j - (M - D_i) \cdot (1 - f_{i',i}^j) \end{aligned} \tag{8}$$

Equation (7) ensures that when multiple candidate windows on the same resource are taken into account in the constraint conditions, the sequence of the two executed tasks can be determined simultaneously without causing a conflict in the observation times. The variable $f_{i,i'}^j = 1$ indicates that the tasks M_i and $M_{i'}$ are arranged to be executed simultaneously with task M_i executed after $M_{i'}$, and the variable $f_{i',i}^j = 1$ indicates that tasks M_i and $M_{i'}$ are arranged to be executed simultaneously with task M_i' executed after M_i. Otherwise, $f_{i,i'} = 0$ and $f_{i',i} = 0$. The introduced binary

Fig. 2 Schematic of multiple coverage window calculation



variables must also satisfy the following conditions:

$$\begin{cases} f_{i,i'}^{j}+f_{i',i}^{j}\le \sum_{k\in 1,2,\ldots,N_{i,j}} x_{i,j}^{k} \\ f_{i,i'}^{j}+f_{i',i}^{j}\le \sum_{k'\in 1,2,\ldots,N_{i,j}} x_{i',j}^{k'} \end{cases} \tag{9}$$

Moreover, since each task is considered for execution at most once during the entire simulation period, the introduced variables $f_{i,i'}^{j}$ and $f_{i',i}^{j}$ must also satisfy the condition:

$$\sum_{R_i\in R(M_i)\cap R(M_{i'})}\left(f_{i,i'}^{j}+f_{i',i}^{j}\right)\le 1 \tag{10}$$

(5) Constraints on the values of optimization variables. For any $M_i\in M$, resource $R_j\in R(M_i)$,and all $k\in\{1,2,\ldots,N_{i,j}\}$, the following constraints apply:

$$x_{i,j}^{k}\in\{0,1\} \tag{11}$$

4 Design of Indicators for Task Selection and Observation Time Window

Based on the formal description of symbols, variables, objectives, and constraints for the multi-satellite cooperative scheduling problem as previously discussed, this section analyzes the degree of flexibility or busyness from the perspectives of tasks and resources, respectively. Furthermore, four indicators have been proposed to evaluate the attributes of time windows under tasks and resources.

4.1 Analysis of Visible Time Window Features

4.1.1 Analysis of Task Arrangement Flexibility

Multiple coverage windows refer to a classification of visibility time windows for a certain type, usually categorized based on the same satellite resources or the same mission objectives. These are intersected in time to obtain a set of time windows with overlap. In the problem of multi-satellite cooperative scheduling, the order in which these windows are scheduled significantly influences the quality of the algorithmic outcomes. And the scheduling sequence depends on the objective function and other metrics such as window contention or observational target flexibility. These metrics are often calculated based on the overlap results. Specifically:

(1) From the resource perspective, the degree of overlap in the multiple coverage windows represents the number of tasks that can be observed by the resource at a given moment.
(2) From the task perspective, the degree of overlap in the multiple coverage windows represents the number of resources that can observe the task at a given moment.

As shown in Fig. 2, the calculation of dual and triple coverage windows from the mission perspective is demonstrated.

4.1.2 Analysis of Resource Availability

During the execution of the algorithm, originally visible time windows may become fully or partially unavailable due to their occupation by previously scheduled tasks. Consequently, it is necessary to compute the complement of each task's entire set of visible time windows with all previously scheduled windows to ascertain the actual available time windows.

As shown in Fig. 3, for all the visible time Windows of a task, complement operations should be performed with all the scheduled time Windows on the same resource of the window to obtain the real visible time window.

In the specific process of task scheduling, the complement operation must also take into account the transition times between the current task and the subsequently scheduled tasks, these transition times need to be subtracted as well. The situation is shown in Fig. 4. The transition time between the preceding tasks needs to be considered according to the situation in determining the execution time stage.

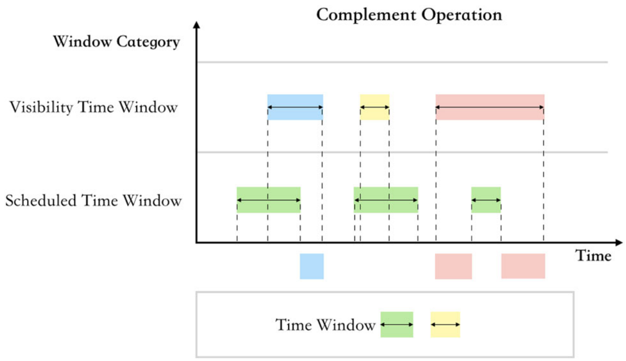


Fig. 3 Complement operation schematic

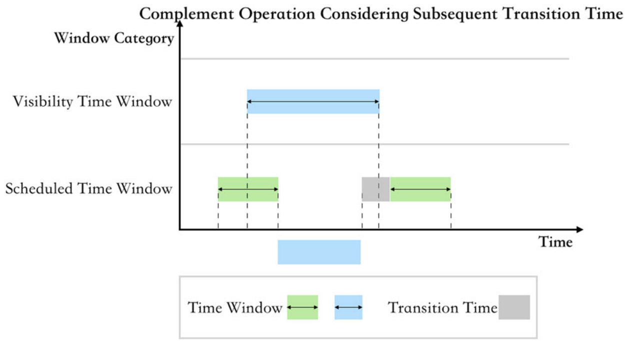


Fig. 4 Schematic of complement operation considering subsequent transition time

4.2 Design of Selection Metrics for the Resource and Corresponding Observation Window

4.2.1 Impact Degree

The impact degree is a metric used to assess the degree of hindrance a scheduled task or observation window poses to the scheduling of other tasks or windows. $BI^k_{i,j}$ denotes the impact degree of the kth visible window of ground task M_i on sensor R_j. BI_i represents the impact degree for task M_i.

For observation windows:

$$\begin{aligned} itLength^{xjz}_{ijk} &= min(End^k_{i,j} + tran_j, End^z_{x,j} + tran_j) \\ &\quad - max(Start^k_{i,j} - tran_j, Start^z_{x,j} - tran_j) \\ &\quad \times (x \neq i) \end{aligned} \tag{12}$$

$$BI^k_{i,j} = \sum_{tw^z_{x,j} \in TW \cap x \neq i} \left(\frac{itLength^{xjz}_{ijk}}{End^k_{i,j} - Start^k_{i,j}} \times w_x \right) \tag{13}$$

For tasks:

$$BI_i = \sum_{tw^k_{i,j} \in TW} BI^k_{i,j} \tag{14}$$

In this context, $itLength^{xjz}_{ijk}$ refers to the actual duration of overlap between the kth visible window for ground task M_i on sensor R_j and all other visible windows on the same sensor, taking into account the transition times before and after. The outcome is utilized as a weight for other tasks on the sensor, which when multiplied by the task profit, yields the impact degree of the observation window.

The impact degree of task BI_i is derived by summing the impact degrees of all visible windows associated with task i.

4.2.2 Conflict Degree

The conflict degree is an indicator used to evaluate the level of contention for time windows. $CFLCT^k_{i,j}$ denotes the contention degree for the kth time window of resource R_j allocated to task M_i. The conflict degree is computed by:

$$\begin{aligned} itLeng^{jz}_{ijk} &= min(End^k_{i,j}, UEnd^z_j) \\ &\quad - max(Start^k_{i,j}, UStart^z_j) \\ CFLCT^k_{i,j} &= \sum_{utw^z_j \in UTW_j} itLeng^{jz}_{ijk} \times multi^z_j \end{aligned} \tag{15}$$

Assuming a window is fully scheduled, the intersection of the union set of windows from the resource perspective is sequentially calculated. The length of the intersection, denoted as $itLeng^{jz}_{ijk}$, is multiplied by the degree of overlap of the union window set, termed $multi^z_j$, and the results are summed to derive the window's conflict degree.Where $UEnd$ is the minimum possible end time for other tasks on that resource and $UStart$ is the latest possible start time for other tasks on that resource.

4.2.3 Flexibility

Flexibility is an indicator that measures the number of usable time windows for a task. $Flex_i$ denotes the flexibility of task M_i.

$$Flex_i = \sum_{tw^z_{x,y} \in TW \cap x=i} count(End^z_{x,y} - Start^z_{x,y} \geq D_x) \tag{16}$$

Flexibility is calculated by counting the number of visible time windows for the task. Each counted window must have a duration greater than or equal to the required minimum observation time.

4.2.4 Bad Impact

Bad impact is a comprehensive assessment of task scheduling, where I_i denotes the bad impact for ground task M_i.

$$I_i = \frac{BI_i \times Flex_i}{w_i} \tag{17}$$

A lower impact degree indicates that scheduling the task causes fewer bad side effects on the scheduling of other tasks. Less flexibility signifies that the task is more susceptible to being influenced by tasks previously scheduled, thus making it more likely that its limited available time windows will be occupied. The bad impact metric is derived by multiplying the task's Impact Degree with its flexibility, and then dividing by the weight. A lower value of this metric indicates a higher priority for task consideration.

This metric, which comprehensively considers weights, available time windows, and their impacts on other time windows, serves as an integrated indicator developed through the collection and analysis of global information. It offers significant guidance in task scheduling.

5 Variable Neighborhood Tabu Search Algorithm

Based on the multi-satellite cooperative scheduling model and the task and observation window selection metrics analyzed under the characteristics of task visible time windows, this chapter further designs an improved variable neighborhood tabu search algorithm to solve the multi-satellite cooperative scheduling problem. Initially, the paper implements an initial solution constructed through various greedy methods, designs task scheduling strategies using multiple rule-based heuristic methods, and proposes three neighborhood structures: insertion, exchange with re-arrangement, and deletion. Moreover, two types of tabu list structures are designed: one based on the characteristics of individual time windows and the other on the characteristics of the solution itself. Additionally, the algorithm employs a weighted randomness method to escape local optima and to incorporate randomness globally or locally when arranging tasks, thus preventing the solution process from being trapped in local optima.

5.1 Initial Solution Construction Based on Greedy Rules

5.1.1 First-Come, First-Served Greedy Algorithm

The execution steps of the greedy algorithm based on first-come-first-served design in this paper are shown as follows.

Step 1: Select the scenario and read the information on tasks and time windows.

Step 2: Calculate the coverage multiplicity, conflict degree, and flexibility for each task.

Step 3: Sort all visible time windows (TWS) within the scenario by start time, from earliest to latest.

Step 4: Enter a loop and check; if the index equals the length of TWS, proceed to the next step; otherwise, go to Step 4-1.

Step 4-1: Determine whether the task corresponding to TWS has been observed. If it has, go to Step 4-2; if not, return to Step 4-2 and increment the index by 1.

Step 4-2: Execute the earliest arrangement algorithm.

Step 5: End.

5.1.2 Minimum Bad Impact Greedy Algorithm

The greedy algorithm based on minimum bad impact(Min_BI) designed in this paper is different from the greedy algorithm based on first-come-first-served in that it employs the ranking evaluation metric of minimum bad impact to rank the tasks.

5.2 Rule-Based Heuristic Strategies

5.2.1 Far-Left Scheduling Algorithm

In the context of multi-satellite cooperative scheduling, once an available time window for task observation has been selected, it is necessary to evaluate whether this window can accommodate the minimum required observation time for the task and the transition times between adjacent tasks, and then select an appropriate segment of the window for task execution.

While the timing algorithm used for deciding the execution times of tasks in the exchange and re-arrangement neighborhood of the tabu search algorithm slightly differs, all algorithms and neighborhood structures discussed in this paper utilize the same execution timing determination algorithm after identifying the insertion window for a task, referred to as the far-left scheduling algorithm.

The algorithm is described as follows. Initially, a set complement operation is performed between the visible observation windows and the already scheduled windows on the same sensor, followed by a sequential traversal of the resulting time windows TW. The first step is to verify if the length of TW meets the minimum observation duration required for the task. If it does, the task window is scheduled in one of three scenarios. Since the transition duration with subsequent tasks has already been accounted for in the set complement operation, it is only necessary to arrange the transition duration and execution time based on the gap between TW and the preceding tasks.

The following strategies are employed for scheduling time:

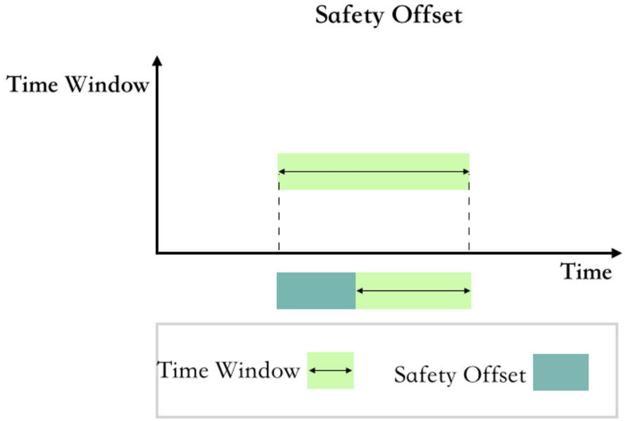


Fig. 5 Safety offset schematic

(1) If there is no already scheduled window LTW to the left of TW, the transition duration need not be considered, and the task is scheduled from the far-left side of TW.
(2) If there is an already scheduled window LTW to the left of TW and the gap between TW and LTW is equal to or greater than the transition duration, the task is similarly scheduled from the far-left side of TW.
(3) If the gap between TW and LTW is less than the transition duration, it implies that part of TW on the left must be used for the transition period, with the task being scheduled from the far-left side immediately after the transition period from LTW.

Employing such a strategy maximizes the use of visible time windows for observations, placing the transition periods in idle times, thereby enhancing the utilization rate of time windows.

When inserting tasks, a random execution time selection algorithm is also considered. Based on the three aforementioned strategies, a safety offset is calculated, and a start time is randomly selected within this offset range for observation, as shown in Fig. 5.

The risk associated with this random strategy is that it may lead to fragmented idle windows being unused, thereby reducing the overall utilization rate of visible time windows. However, it helps to avoid falling into local optimum solutions.

5.2.2 Absolute Optimal Time Windows

There exists a type of visible time window that, even with maximized consideration for transition times, does not intersect with any other visible windows of the same resource. This means that scheduling this time window does not badly impact the scheduling of other tasks, and the length of the window meets the minimum observation time required for the associated task. Such time windows are referred to as absolute optimal time windows, as illustrated in Fig. 6.

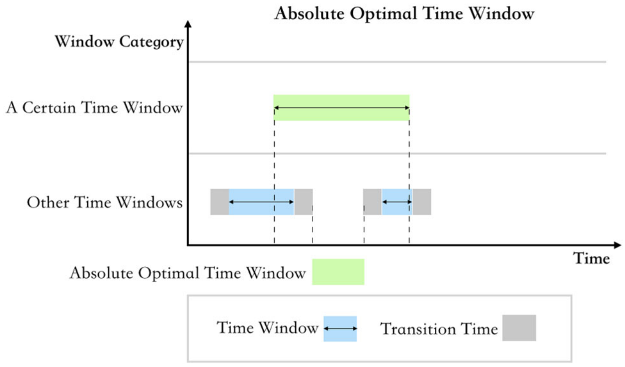


Fig. 6 Absolute optimal time window

During the initial phase of the tabu algorithm, a priority is placed on identifying and scheduling all absolute optimal time windows. Throughout the course of the algorithm, these absolute optimal time windows are protected, and they are not deleted, replaced, or considered for the rescheduling of tasks to other windows.

5.2.3 Weighted Randomness

In the execution of the algorithm, it is necessary to incorporate some randomness to explore broader search domains and prevent stagnation in local optima. For instance, when selecting tasks for insertion, one might initially attempt to arrange tasks based on decreasing flexibility or increasing bad impact. However, this approach could lead to a unidirectional search pattern, potentially missing intermediate solutions that could lead to the optimal result. A completely random selection method, on the other hand, would result in inefficient and unpredictable search outcomes. Therefore, this paper employs a weighted random number algorithm that balances randomness with directional searching needs.

Given an integer P, a weighted random number generation algorithm with greater weight closer to zero is used, where P is determined by the number of tasks.

Algorithm 1 Weighted Random Selection

```
Generate a random number $randNum$, uniformly distributed between 0 and $P$
Calculate $invRandNum = 1.0/(randNum + 1)$
Generate a second random number $randProb$, uniformly distributed between 0.0 and 1.0
if $randProb > invRandNum$ then
    Reject $randNum$ and regenerate it
end if
```

5.3 Neighborhoods Design

5.3.1 Insertion Neighborhood

The insert neighborhood operation attempts to insert a new task arrangement into the current solution, with the following process:

Step 1: Sort the task list in ascending order according to flexibility.
Step 2: Select a task with a weighted random probability. If the task is already scheduled, return to $Step2$ to re-select a task; otherwise, proceed to the next step.
Step 3: Perform the complement operation on all visible time windows of the task with all scheduled time windows to obtain available time windows, and sort these available time windows according to the least bad impact.
Step 4: Schedule the task using the leftmost scheduling algorithm. If the task is successfully scheduled, move to the next step; otherwise, select the next time window and re- execute the far-left scheduling algorithm.
Step 5: If the scheduled time window is in the taboo list, select the next time window and return to $Step4$; otherwise, continue to the next operation.
Step 6: Insert this new window into the current solution to generate a new neighbor, and the algorithm terminates.

5.3.2 Exchange and Rearrangement Neighborhood

The exchange and rearrangement neighborhood attempts to replace an already scheduled task with a new task and tries to rearrange the replaced task to a different position, the steps are as follows.

Step 1: Deleting the old task. Traverse the global tasks sorted by decreasing flexibility, selecting a task to be replaced through weighted random probability in each iteration. If the selected task has been observed and its corresponding window is not an absolute optimal time window, then delete this scheduled task from the current solution.
Step 2: Inserting a new task. Search for a new task to insert in ascending order of flexibility, ensuring that the task has not been observed and is not the old task. Perform a set complement operation on all visible time windows of the new task to obtain feasible time windows, and sort these by increasing bad impact. Traverse the feasible windows of the new task, ensuring that these windows overlap with the deleted task's windows to achieve an 'exchange' effect, and apply the far-left scheduling algorithm. If not on the tabu list, insert the new task into the current solution.
Step 3: Perform the complement operation on all visible time windows of the task with all scheduled time windows to obtain available time windows, and sort these available time windows according to the least bad impact.
Step 4: Rearranging the old task. Traverse the visible time windows of the deleted task, considering new time windows that do not overlap with the deleted execution window to achieve an *solution movement effect*. If there is no overlap, apply the far-left scheduling algorithm, and upon successful arrangement, insert the rearranged old task's execution window into the current solution.

5.3.3 Deletion Neighborhood

The delete neighborhood operation aims to try removing time windows from the current solution. The steps are as follows:

Step 1: Sort the already scheduled windows ($ATWS$) in descending order of bad impact, and select a task with greater bad impact through weighted random probability.
Step 2: If the time window corresponding to the task is in the taboo list, or is the absolute optimal time window, then return an empty neighbor and skip to $Step4$; otherwise, proceed to the next step.
Step 3: Delete that window and generate a neighbor.
Step 4: The algorithm terminates.

5.4 Implementation Process of the Algorithm

This paper presents a variable neighborhood tabu search algorithm applied to multi-satellite cooperative scheduling. The algorithm treats the collection of already scheduled time windows as solutions and neighbors, and designs three types of neighborhood structures: insertion, exchange and rearrangement, and deletion. It utilizes two types of tabu lists: one based on the characteristics of the scheduled time windows and the sequential order of tasks scheduled on the same sensor, with the tabu list length being adaptively variable. The algorithm also employs strategies such as counting unimproved iterations and random escaping to avoid local optima.

The process of tabu search algorithm designed in this paper is as follows.

Step 1: Select a scenario and read the information on tasks and time windows.
Step 2: Calculate the coverage multiplicity, degree of conflict, and flexibility for each task.

Step 3: Perform the greedy algorithm with the least bad impact, and select all or part of the schedule results of the greedy algorithm as the initial solution.

Step 4: Sequentially explore the insert neighborhood operation, explore the exchange and reschedule neighborhood operation, and explore the delete neighborhood operation.

Step 5: Update the current solution.

Step 5-1: Identify the optimal neighbor based on the objective function, update the current solution to the optimal neighbor, and if this neighbor is better than the global optimal neighbor, then update the global optimal neighbor.

Step 5-2: If no improvement is made after a specified number of iterations, randomly select one from all neighbors as the current solution and add the current solution to the taboo list. If the length of the taboo list exceeds the specified length, remove the element that entered the taboo list earliest.

Step 6: If the specified number of iterations for the algorithm has been reached, then the algorithm ends; otherwise, go back to $Step4$ and continue the neighborhood exploration operation.

6 Experimental Analysis

6.1 Dataset

The dataset used in this paper consists of a group of public datasets, which are classified based on periodic length into two categories: 24 h and 48 h. They are further divided into two categories based on the number of available resources and into five categories based on the number of ground targets, as shown in Table 1. The term ‘available resources’ refers to the total number of available sensors across all satellites in the scenario. The ‘number of ground targets’ refers to the number of tasks that have potential visible time windows.

The minimum observation duration for ground targets is a random integer in the interval [3, 10], and the observation profit for ground targets is a random integer in the interval [1, 10]. It is assumed that the simulation periods begins at 2023-01-01 00:00:00 and has a length of either 24 h or 48 h.

Each dataset provides detailed information about scenarios, satellites, sensors, and targets, It is generated with CSTK (China Satellite Tool Kit), which is developed by the Spatial Information Engineering Team, China University of Geosciences (Wuhan). The scenarios contain multiple satellites, tasks, and visible time windows that describe all the information necessary for scheduling and planning tasks, including the scenario name, the earliest start time, and the latest end time, as well as satellites, tasks, and visible time windows. The visible time window labels include the satellite name, sensor name, the observed task name, and the start and end times. Task labels include the name of the ground target, the required minimum observation duration, the profit obtained from the observation task, the sensor type needed for the observation, and the earliest start and latest end times of the task. Satellite labels include the satellite name and multiple sensors. Sensor labels further include the sensor name, the type of coverage provided by the sensor, the minimum transition duration required between two adjacent tasks on the sensor, and whether the sensor is operational.

Table 1 Characteristics of the dataset

Number	Simulation period length (h)	Available resource number	Number of ground targets
1	24	3	100
2	24	3	200
3	24	3	300
4	24	5	100
5	24	5	200
6	24	5	300
7	24	5	400
8	24	5	500
9	48	5	100
10	48	5	200
11	48	5	300
12	48	5	400
13	48	5	500

The dataset provides theoretical optimal values for the objective functions, which are the total profits and total number of completed observation tasks, as shown in Table 2. The maximum profits refers to the upper boundary of profits that can be achieved by a dataset. The maximum number of tasks refers to the upper boundary of the number of observation tasks that can be achieved by a dataset.

Using the optimal dataset allows for an intuitive evaluation of the algorithm’s results, particularly the gap relative to the best results can reflect the algorithm’s capability to escape better solutions and achieve the optimal.

6.2 Results of the Greedy Algorithm

Tables 3 and 6 presents the scheduling results of the greedy algorithms based on FIFO and Min_BI for the dataset selected in this paper. Within the FIFO-based greedy algorithm, the scheduling results of 8 datasets have a gap within 20% of the maximum profit, and the gap between the FIFO scheduling results and the maximum profit is not less than 10% for any case, with the maximum number of tasks com-

Table 2 Theoretical optimal value of the dataset

Number	Max profits	Max number of tasks scheduled
1	194	26
2	599	82
3	629	85
4	318	44
5	888	129
6	928	131
7	1198	163
8	1348	179
9	495	72
10	1042	155
11	1226	176
12	1787	265
13	2034	290

Table 3 Comparison of scheduling profit results for greedy algorithms

Number	Max profits	FIFO		Min_BI	
		Scheduling result	Gap	Scheduling result	Gap
1	194	153	21.1%	188	3.1%
2	599	508	15.2%	534	10.9%
3	629	505	19.7%	573	8.9%
4	318	252	20.8%	307	3.5%
5	888	755	15.0%	806	9.2%
6	928	747	19.5%	813	12.4%
7	1198	940	21.5%	975	18.6%
8	1348	1032	23.4%	1081	19.8%
9	495	435	12.1%	467	5.7%
10	1042	922	11.5%	940	9.8%
11	1226	1015	17.2%	1072	12.6%
12	1787	1421	20.5%	1496	16.3%
13	2034	1610	20.8%	1695	16.7%

pleted being within 10% of the optimal solution. On the other hand, the Min_BI-based greedy algorithm shows significant improvement in scheduling results when targeting maximum profit, with all cases having a gap within 20% of the maximum profit and 6 cases having a gap within 10% of the optimal solution. However, when targeting the maximum number of tasks completed, the performance of the Min_BI-based algorithm decreases, with the gap between the scheduling results and the maximum number of tasks completed being not less than 10% for any case (Table 4).

Table 4 Comparison of results on the number of tasks scheduled by the greedy algorithm

Number	Max number of tasks scheduled	FIFO		Min_BI	
		Scheduling result	Gap	Scheduling result	Gap
1	26	24	7.7%	23	11.5%
2	82	79	3.7%	73	11.0%
3	85	82	3.5%	68	20.0%
4	44	40	9.1%	40	9.1%
5	129	122	5.4%	116	10.1%
6	131	123	6.1%	115	12.2%
7	163	157	3.7%	138	15.3%
8	179	171	4.5%	153	14.5%
9	72	68	5.6%	66	8.3%
10	155	146	5.8%	139	10.3%
11	176	163	7.4%	155	11.9%
12	265	243	8.3%	220	17.0%
13	290	270	6.9%	246	15.2%

6.3 Results of the Tabu Search Algorithm

When the objective function is set to maximize profits, the tabu search algorithm is run using the superior results of the Min_BI-based greedy algorithm as the initial solution, with the scheduling results presented in Tables 5 and 7. For all cases, the variable neighborhood tabu Search algorithm proposed in this paper achieves results within a 10% gap from the optimal solution, and for 4 cases, the gap to the optimal solution is merely within 5%, indicating an improvement over the results of the greedy algorithm and the traditional tabu search algorithm.

When the objective function is to maximize the number of tasks completed, the experimental results are shown in Tables 6 and 8. Considering the FIFO-based greedy algorithm as a comparative object, the variable neighborhood tabu search algorithm also achieves the best results, with the gap to the optimal solution for all cases within 5% (Figs. 7, 8).

7 Conclusion

This paper studied the flexibility features and availability of visible time windows, and described the significance of union or complement operations considering the post-transition duration. The proposed indicators of the impact degree, the conflict degree, the flexibility, and the bad impact can effectively evaluate the features of resources and corresponding time windows. For the focus window selection issue, this paper has designed a leftmost arrangement algorithm to select the start execution time after confirming the specific window

Table 5 Comparison of scheduling profit results for multiple algorithms

Number	Max profits	Min_BI	Traditional tabu search	Variable neighborhood tabu search
1	194	188	190	190
2	599	534	563	575
3	629	573	591	597
4	318	307	311	311
5	888	806	815	820
6	928	813	826	850
7	1198	975	1054	1089
8	1348	1081	1189	1236
9	495	467	470	473
10	1042	940	962	973
11	1226	1072	1108	1121
12	1787	1496	1577	1627
13	2034	1695	1836	1836

Table 6 Comparison of the number of task scheduling results for multiple algorithms

Number	Max number of tasks scheduled	Min_BI	Traditional tabu search	Variable neighborhood tabu search
1	23	23	24	25
2	73	73	79	79
3	68	68	82	83
4	40	40	41	42
5	116	116	122	124
6	115	115	124	127
7	138	138	159	159
8	153	153	172	173
9	66	66	70	70
10	139	139	148	151
11	155	155	166	171
12	220	220	245	258
13	246	246	273	278

Table 7 Comparison of the gap in task scheduling profits results of multiple algorithms to maximum profits

Number	FIFO	Min_BI	Traditional tabu search	Variable neighborhood tabu search
1	21.1%	3.1%	2.1%	2.1%
2	15.2%	10.9%	6.0%	4.0%
3	19.7%	8.9%	6.0%	5.1%
4	20.8%	3.5%	2.2%	2.2%
5	15.0%	9.2%	8.2%	7.7%
6	19.5%	12.4%	11.0%	8.4%
7	21.5%	18.6%	12.0%	9.1%
8	23.4%	19.8%	11.8%	8.3%
9	12.1%	5.7%	5.1%	4.4%
10	11.5%	9.8%	7.7%	6.6%
11	17.2%	12.6%	9.6%	8.6%
12	20.5%	16.3%	11.8%	9.0%
13	20.8%	16.7%	9.7%	9.7%

Table 8 Comparison of the gap in the number of tasks scheduled by multiple algorithms to the maximum number of tasks scheduled

Number	FIFO	Min_BI	Traditional tabu search	Variable neighborhood tabu search
1	7.7%	11.5%	7.7%	3.8%
2	3.7%	11.0%	3.7%	3.7%
3	3.5%	20.0%	3.5%	2.4%
4	9.1%	9.1%	6.8%	4.5%
5	5.4%	10.1%	5.4%	3.9%
6	6.1%	12.2%	5.3%	3.1%
7	3.7%	15.3%	2.5%	2.5%
8	4.5%	14.5%	3.9%	3.4%
9	5.6%	8.3%	2.8%	2.8%
10	5.8%	10.3%	4.5%	2.6%
11	7.4%	11.9%	5.7%	2.8%
12	8.3%	17.0%	7.5%	2.6%
13	6.9%	15.2%	5.9%	4.1%

for observation. The concept of the absolute optimal time window has been proposed, which, once marked, remains unchanged throughout the algorithm execution, serving to reduce the solution space and improve the quality of the solution. The concept of weighted randomness is also introduced for the selection of resources and execution time windows with a tendency towards randomness while simultaneously ensuring the controllability of the exploration direction.

This paper has solved datasets of various sizes through an improved tabu search algorithm, focusing on the variable neighborhood structure of the tabu search algorithm to solve the multi-satellite collaborative scheduling problem, designing insertion, deletion, and swapping and rearranging three types of neighborhood results for exploring neighbors. The tabu list for the sequence of internal sensor observation tasks uses a variable-length design, positively correlated with the current solution count, thus adaptively solving different task or window scales. Experimental results validate the effectiveness and efficiency of the proposed method.

Based on the above content, further research in the following areas is still to be conducted: (1) When solving for better

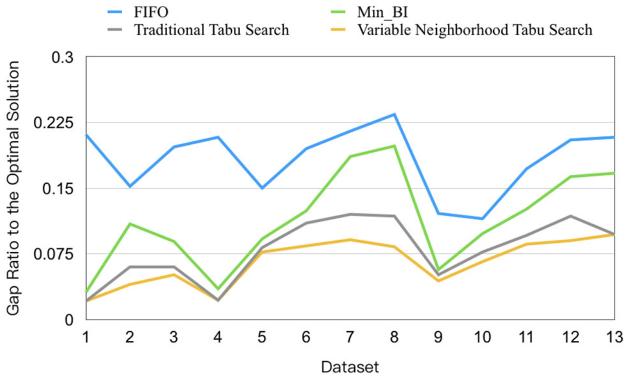


Fig. 7 Gap ratio to the optimal solution

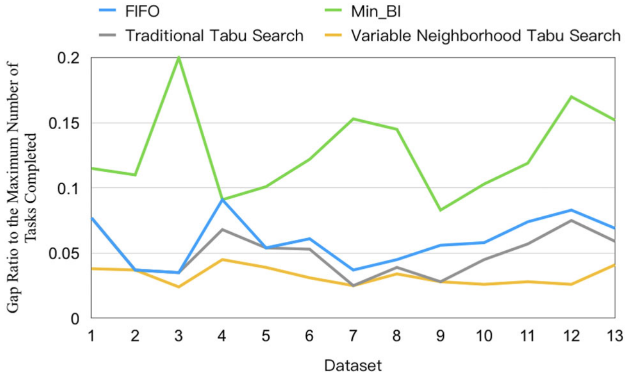


Fig. 8 Gap ratio to the maximum number of tasks completed

solutions in larger datasets, the solving speed is slow. (2) The maximum number of iterations allowed without improvement has not been set to a variable adaptive length, which cannot effectively handle datasets of different magnitudes, limiting the direction of the solution.

Author Contributions Xiaoyu Chen, Qi Gao: performed the data analyses and wrote the manuscript; Shengren Peng, Shihui Song, Yufei Liu: performed the experiment; Guangming Dai: contributed to the conception of the study; Maocai Wang: contributed significantly to analysis and manuscript preparation; Changxing Zhang: helped perform the analysis with constructive discussions.

Funding This letter is funded by the National Natural Science Foundation of China (42271391; 62006214), Special Project of Hubei Key Research and Development Program (2023BIB015), Joint Funds of Equipment Pre-Research and Ministry of Education of China (8091B022148), the 14th Five-year Pre-research Project of Civil Aerospace in China, and Hubei excellent young and middle-aged science and technology innovation team plan project (T2021031).

Data Availibility Statement The authors confirm that the data supporting the findings of this study are available within the article.

Declarations

Ethical approval Al procedures performed in studies involving human participants were in accordance with the ethical standards of the institutional and/or national research committee and with the 1964 Helsinki declaration and its later amendments or comparable ethical standards.

Informed consent Informed consent was obtained from all individual participants included in the study.

Conflict of interest The authors declare no competing financial or non-financial interests.

References

1. Wang J (2007) Research on comprehensive mission scheduling model and optimization methods for imaging satellites. PhD thesis, National University of Defense Technology
2. Decuyper M, Chávez RO, Lohbeck M, Lastra JA, Tsendbazar N, Hackländer J, Herold M, Vågen T-G (2022) Continuous monitoring of forest change dynamics with satellite time series. Remote Sens Environ 269:112829
3. Lyman JM, Johnson GC (2023) Global high-resolution random forest regression maps of ocean heat content anomalies using in situ and satellite data. J Atmos Ocean Technol 40(5):575–586
4. Zhao Z, Shang H, Wei B (2022) Tackling nonconvex collision avoidance constraints for optimal trajectory planning using saturation functions. J Guid Control Dyn 45(6):1002–1016
5. Zhao Z, Shang H (2023) Trajectory planning for spacecraft formation reconfiguration using saturation function and difference-of-convex decomposition. IEEE Trans Aerosp Electron Syst 60(1):857–866
6. Zhao Z, Shang H, Dong Y (2022) Autonomous attitude planning for gravity wave detection using hybrid convex optimization. Aerosp Sci Technol 130:107923
7. Chen X, Reinelt G, Dai G, Wang M (2018) Priority-based and conflict-avoidance heuristics for multi-satellite scheduling. Appl Soft Comput 69:177–191
8. Chen X, Reinelt G, Dai G, Spitz A (2019) A mixed integer linear programming model for multi-satellite scheduling. Eur J Oper Res 275(2):694–707
9. Chen X, Song Z, Dai G, Wang M, Ortore E, Circi C (2021) A general formal method for manifold coverage analysis of satellite constellations. IEEE Trans Aerosp Electron Syst 58(2):1462–1479
10. Chen X, Gu W, Dai G, Xing L, Tian T, Luo W, Cheng S, Zhou M (2024) Data-driven collaborative scheduling method for multi-satellite data-transmission. Tsinghua Sci Technol 29(5):1–18
11. Glover F (1986) Future paths for integer programming and links to artificial intelligence. Comput Oper Res 13(5):533–549
12. Glover F (1989) Tabu search-part i. ORSA J Comput 1(3):190–206
13. Li D, Wang M et al (1997) Tabu search algorithm for the traveling salesman problem with precedence relations. J Northeast Univ 06:77–80
14. Cheng Hao L, Yan Hong W, Wei QJ, Zhao DW, Rui S (2020) Product service scheduling problem with service matching based on Tabu search method. J Adv Transp 2020:1–9
15. Chen C, Hu D et al (2007) Application of genetic algorithms (ga) and tabu search algorithms (ts) in location and routing problems of distribution centers. Syst Eng Theory Pract 27(09):171–176
16. Kang Y, Huang W (2004) Heuristic algorithm based on tabu search for solving the circular packing problem. Journal of Computer Research and Development 09:1554–1558
17. Nunes JUN, Bretas AS (2017) Voltage regulators allocation in power distribution networks: a tabu search approach. In: 2017 19th international conference on intelligent system application to power systems (ISAP), pp 1–6. IEEE

18. Ziegler DU, Prettico G, Mateo C, San Román TG (2023) Methodology for integrating flexibility into realistic large-scale distribution network planning using tabu search. Int J Electr Power Energy Syst 152:109201
19. Bosisio A, Berizzi A, Lupis D, Morotti A, Iannarelli G, Greco B (2022) A tabu-search-based algorithm for distribution network restoration to improve reliability and resiliency. J Mod Power Syst Clean Energy 11(1):302–311
20. Awad FH, Al-Kubaisi A, Mahmood M (2022) Large-scale timetabling problems with adaptive tabu search. J Intell Syst 31(1):168–176
21. Saleh HA, Fidanova S (2003) Tabu search procedures for optimising global positioning system surveying networks
22. Rahdar S, Ghanbari R, Ghorbani-Moghadam K (2022) Tabu search and variable neighborhood search algorithms for solving interval bus terminal location problem. Appl Soft Comput 116:108367
23. Bianchessi N, Righini G (2008) Planning and scheduling algorithms for the Cosmo–Skymed constellation. Aerosp Sci Technol 12(7):535–544
24. Zuo C, Wang H (2010) Research on geodetic satellite mission scheduling based on tabu search algorithm. Comput Eng Appl 46(1):215-217
25. Li J, He R, Yao F et al (2013) Variable neighborhood tabu search algorithm for integrated scheduling of imaging satellites. Syst Eng Theory Pract 33(12):3040–3044
26. Yang W, He L, Liu X, Meng W, Chen Y (2023) A fast insertion tabu search with conflict-avoidance heuristic for the multisatellite multimode crosslink scheduling problem. Tsinghua Sci Technol 29(3):843–862