

Article

A Hybrid Genetic Algorithm for Ground Station Scheduling Problems

Longzeng Xu, Changhong Yu *, Bin Wu and Ming Gao

School of Information and Electronic Engineering, Zhejiang Gongshang University, Hangzhou 310018, China; 21020090001@pop.zjgsu.edu.cn (L.X.); 21020090010@pop.zjgsu.edu.cn (B.W.); gaoming@zjgsu.edu.cn (M.G.)

* Correspondence: yuch@mail.zjgsu.edu.cn

Abstract: In recent years, the substantial growth in satellite data transmission tasks and volume, coupled with the limited availability of ground station hardware resources, has exacerbated conflicts among missions and rendered traditional scheduling algorithms inadequate. To address this challenge, this paper introduces an improved tabu genetic hybrid algorithm (ITGA) integrated with heuristic rules for the first time. Firstly, a constraint satisfaction model for satellite data transmission tasks is established, considering multiple factors such as task execution windows, satellite-ground visibility, and ground station capabilities. Leveraging heuristic rules, an initial population of high-fitness chromosomes is selected for iterative refinement. Secondly, the proposed hybrid algorithm iteratively evolves this population towards optimal solutions. Finally, the scheduling plan with the highest fitness value is selected as the best strategy. Comparative simulation experimental results demonstrate that, across four distinct scenarios, our algorithm achieves improvements in the average task success rate ranging from 1.5% to 19.8% compared to alternative methods. Moreover, it reduces the average algorithm execution time by 0.5 s to 28.46 s and enhances algorithm stability by 0.8% to 27.7%. This research contributes a novel approach to the efficient scheduling of satellite data transmission tasks.

Keywords: satellite data transmission; genetic algorithm; constraint satisfaction model; tabu search algorithm; heuristic rules



Citation: Xu, L.; Yu, C.; Wu, B.; Gao, M. A Hybrid Genetic Algorithm for Ground Station Scheduling Problems. *Appl. Sci.* **2024**, *14*, 5045. <https://doi.org/10.3390/app14125045>

Academic Editor: Jérôme Morio

Received: 9 April 2024

Revised: 3 June 2024

Accepted: 4 June 2024

Published: 10 June 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In recent years, technological advancements in aerospace engineering have led to a rapid increase in the number of remote sensing satellites. This expansion has resulted in a substantial surge in data transmission tasks and the associated volume of data [1], straining the existing ground station infrastructure beyond its capabilities. Despite the surge in satellite data, the hardware resources of most ground stations have not undergone commensurate upgrades, thus exposing the shortcomings of traditional scheduling methods in managing burgeoning resource allocation demands. These conventional approaches are increasingly inadequate, frequently failing to efficiently allocate finite hardware resources, which gives rise to significant bottlenecks and impedes timely data reception. Consequently, there is an imperative need to develop and implement innovative strategies that enhance data reception processes considering the existing hardware resource limitations [2]. Tackling this challenge is pivotal to improving the overall efficacy and dependability of satellite data transmission systems.

The scheduling of ground station hardware resources is an NP-hard problem [3]. Approaches to its solution include heuristic algorithms, intelligent search algorithms, and hybrid algorithms. Some scholars, utilizing a vast array of effective information derived from accumulated antenna scheduling data from current remote sensing satellite ground stations, have proposed an intelligent scheduling algorithm that synergizes Long Short-Term Memory neural networks with heuristic rules [4]. Others have introduced an

improved Ant Colony Optimization algorithm with dynamically adjustable parameters, specifically targeting the task and resource conflicts arising from complex satellite data transmission scenarios [5]. Moreover, researchers have considered decomposing the multi-ground station resource scheduling problem into several individual ground station resource scheduling issues, incorporating constraints related to antennas and recording equipment, and have suggested an enhanced algorithm that combines heuristic methods with Lagrange relaxation techniques [6]. While the methods investigated by these scholars can, to a certain extent, address the problems of ground station resource scheduling, they predominantly consider only antennas or recording equipment. In practice, however, not only do these components come into play, but hardware resources such as frequency converters and demodulators, along with their corresponding constraints, are also involved. If the constraints tied to ground station resources are fully taken into account, the aforementioned methods are unable to yield the optimal solution to the ground station hardware resource scheduling conundrum.

The genetic algorithm (GA) has the ability to customize coding modes, fitness functions, individual selection mechanisms, crossover strategies, and mutation strategies according to problem characteristics [7], and has excellent global search capability [8], which can effectively deal with the resource scheduling problem of ground stations [9,10].

In the current research on the GA for ground station resource scheduling, limited attention has been paid to incorporating diverse hardware resource constraints. Instead, studies focusing on single-type hardware resource constraints predominate, providing valuable insights. Recognizing the increasing demand for dynamic resource rescheduling in ground stations and the limitations of single-objective optimization, scholars have proposed multi-objective dynamic rescheduling algorithms based on the GA. These algorithms have demonstrated superior performance in terms of optimization and computational overhead compared to existing methods [2].

To address the requirements of theme-oriented satellite data transmission and ensure the timeliness and integrity of thematic data, researchers have introduced an enhanced genetic algorithm integrated with particle swarm optimization. This hybrid approach has proven to be more efficient than existing techniques [11]. Furthermore, considering that resolving satellite data transmission conflicts can provide heuristic information for future satellite revisits, scholars have proposed an improved genetic algorithm that combines immune algorithms. This combined approach is capable of addressing scenarios that prioritize both the efficiency and stability of satellite data transmission scheduling [12]. Collectively, these methodologies highlight the high efficiency of the GA and its potential for integration with other algorithms to address their inherent limitations, further enhancing their performance [13].

However, considering multiple hardware resource constraints will further increase the difficulty of solving the ground station hardware resource scheduling problem. If the satellite data transmission scale also increases synchronously, the solution efficiency of the genetic algorithm will be affected to a certain extent. Some scholars combined the depth-first search (DFS) algorithm and the GA, and fully considered the constraints of the actual ground station receiving system to solve the ground station resource scheduling problem with a single station, multiple antennas, and multiple tasks [14]. Considering the constraints of antennas, inverters, demodulators, and recording devices, DFS is used to improve the local search ability of the genetic algorithm and improve the quality of the solution. Heuristic information is used to optimize the search process of DFS, so that the method can calculate the optimal solution faster. However, as the scale of satellite data transmission gradually increases, DFS is more and more time-consuming, which reduces the solution efficiency of the method. Therefore, hybrid algorithms based on the genetic algorithm dealing with large-scale ground station hardware resource scheduling problems need to be further studied.

In summary, we propose an ITGA combined with heuristic rules. Firstly, the data transmission task set is divided into non-conflicting sub-task sets using a divide-and-

conquer approach. Secondly, initial chromosomes representing the execution order of the data transmission tasks are randomly generated, and the allocation of ground station hardware resources is determined based on this order to calculate the fitness values of these chromosomes. In the ground station's hardware resource scheduling, the algorithm combines a conflict resolution method based on heuristic rules with the ground station's resource allocation method. Data transmission tasks that cannot be executed due to conflicts are pruned in order to enhance both the overall quality of the initial population and individual chromosome quality during iterations. By combining the tabu search algorithm (TSA) with the GA, improvements are made to traditional mutation and crossover operators, preventing repeated searches and avoiding local optimal solutions while considering constraints related to ground station resources. Consequently, an optimal solution for allocating hardware resources in satellite data transmission ground stations is obtained, which has been verified through experiments demonstrating its effectiveness and reliability.

The main contributions of this paper are as follows: A constraint satisfaction model considering multiple ground resources is proposed as a solution model for the hardware resource scheduling problem in satellite data transmission ground stations. An ITGA combined with heuristic rules is introduced to improve the solution efficiency and stability for the aforementioned problem. The superiority of the ITGA is verified through comparative simulation experiments.

The remaining parts of the paper are organized as follows: Section 2 introduces types of ground resources and the receiving system of the ground station and analyzes the satellite data transmission process and transmission conflicts. Section 3 analyzes the constraints and optimization objectives of the hardware resource scheduling problem in satellite data transmission ground stations and, based on this, proposes a satellite data transmission task scheduling model. Section 4 presents the design of the ITGA, along with the designs of the task set partitioning method and the conflict resolution approach. Section 5 validates the effectiveness of the ITGA based on heuristic rules through simulation experiments. Finally, Section 6 concludes the paper.

2. Description of Satellite Data Transmission

2.1. Ground Receiving System

The receiving system of the ground station is an important part of the ground application system, which is cooperated with the satellite tracking subsystem and the data entry subsystem to complete the receiving and recording of the observation data transmitted by the satellite [14]. The antenna of the ground station is responsible for capturing the radio wave signal from the satellite. Through the feed network, the signal of the target frequency band (usually the L- or X-band) is sent to the corresponding processing link, and then amplified to the level that can be detected and processed by the low-noise amplifier (LNA). The received high-frequency signal is converted into a lower-frequency signal by the down converter (DC), and transmitted to the demodulator for demodulation and decoding. Finally, useful information is stored to a hard disk or other storage media through the recording device (RD) in the data processing subsystem [15].

Satellite data reception requires the arrangement of antennas, channel subsystem links, and recording equipment in the data processing subsystem. The channel subsystem links include the first DC (FDC), a LNA, the second DC (SDC), and a demodulator [16]. The three hardware devices are connected by a radio frequency switch matrix (RFSM), intermediate frequency switch matrix (IFSM), and baseband three-stage switch matrix (BTSSM). A typical ground station receiving system structure is shown in Figure 1.

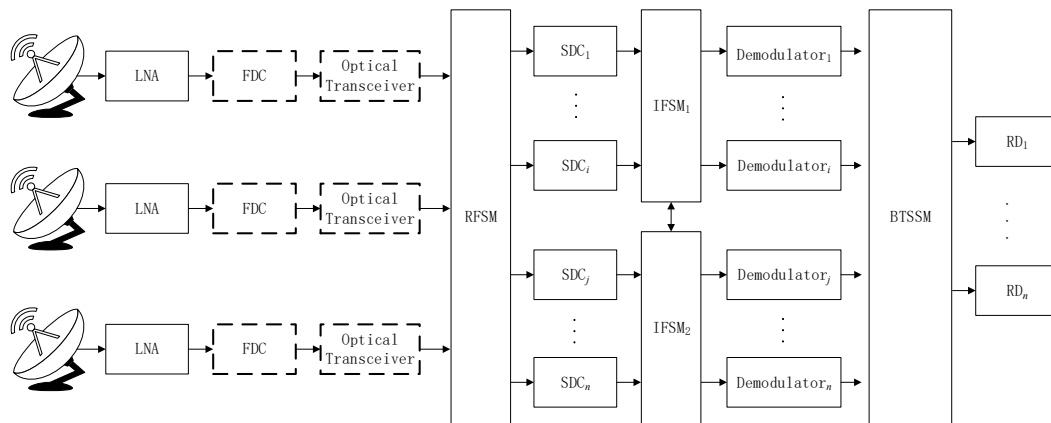


Figure 1. Structure of ground receiving system.

The device represented by the dashed box represents multiple receiving links between the antenna device and the RFSM, and there is a receiving link that does not require the device to perform data reception.

2.2. Conflict in Ground Resource Utilization

During the processing of satellite data transmission tasks at ground stations, three primary types of conflict commonly arise:

- Satellite-to-Ground Visibility Time Conflict (SGVTC);
- Resource Utilization Conflict (RUC);
- Resource Link Conflict (RLC).

When addressing hardware resource scheduling challenges for satellite data transmission tasks at ground stations, it is crucial to mitigate the occurrence of these three conflicts as much as possible. If avoidance is not feasible, effective strategies must be devised to minimize the number of tasks that need to be aborted or delayed due to these conflicts.

2.2.1. SGVTC

SGVTC refers to the conflicts that arise when two or more satellites are present within the acquirable range of the same ground station during a specific time period. As depicted in Figure 2, this can occur when a ground station has satellites 1 through 3 entering its visible range within a particular timeframe, each possessing a distinct visible time window. The figure illustrates that the visible times of satellite 1 and satellite 3 overlap during the $t_1 - t_2$ timeframe, resulting in an SGVTC. The ground station's receiving system must devise strategies to address this conflict or alternatively cancel one of the tasks.

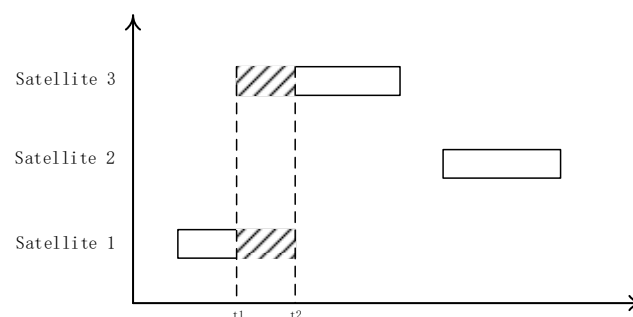


Figure 2. Satellite visible time conflict diagram.

2.2.2. RUC

RUC refers to a situation where two or more satellite data transmission tasks require the use of the same ground station hardware resource during a specific time period. As

illustrated in Figure 3, when satellites 2 and 3 are simultaneously in view, an RUC arises because antenna 2 is already allocated to satellite 1 for data transmission. Consequently, satellites 2 and 3 compete for antenna 1, resulting in a conflict. In such cases, antenna 1 must be assigned to either satellite 2 or satellite 3, while the other task without an allocated antenna must either be assigned to another available antenna or aborted.

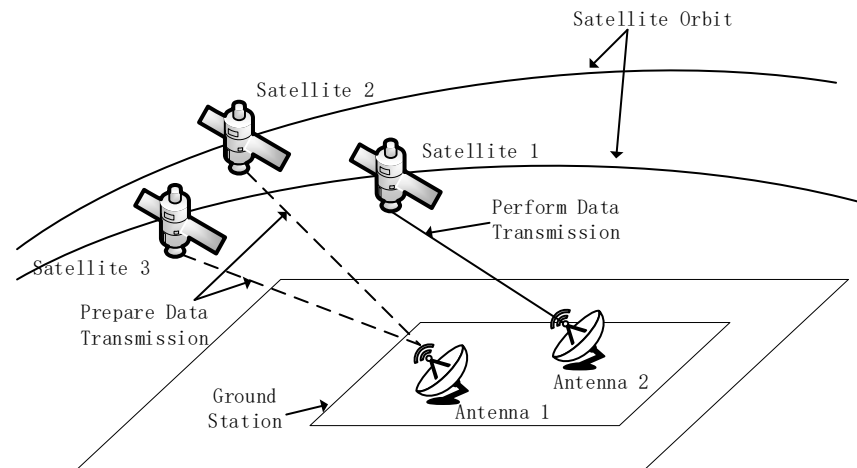


Figure 3. Resource usage conflict diagram.

2.2.3. RLC

RLC occurs when the allocation of equipment for a current task necessitates traversing the IFSM, but the ground station's receiving system has reached its maximum allowed crossings. As shown in Figure 1, assigning SDC 1 through i and demodulators j through n to a task requires the use of a switching path that spans across two IFSMs. If the maximum allowed number of crossings through an IFSM has been reached, the task cannot be accommodated with the intended converters and demodulators. This necessitates either abandoning the task or reassigning it to equipment that does not require matrix crossings.

3. Ground Resource Scheduling Model

To address the hardware resource scheduling problem in satellite data transmission ground stations, it is first necessary to clarify the variables involved in this issue and then establish a ground station resource scheduling model. This model will facilitate a better analysis of the objective function and constraint conditions related to the problem.

Currently, each antenna system in the ground station is equipped with an independent FDC, LNA, and optical transceiver, with essentially consistent parameters. Therefore, we only consider four types of equipment resources: antennas, SDCs, demodulators, and RDs.

3.1. Symbol Definition

To facilitate the description of ground resource scheduling issues, the symbol definitions are first provided, as shown in Table 1.

Table 1. Symbol definitions.

Notations	Definition
AL	List of ground station antennas.
WL	List of visible time windows between satellite and the ground.
T	Task set that converts all satellite-to-ground visible time windows satisfying the user-specified minimum data transmission time into data transmission tasks.
SDCL	List of SDCs at the ground station.
DL	List of demodulators at the ground station.
RDL	List of RDs at the ground station.

Based on the aforementioned symbol definitions, we establish the following ground resource scheduling model for a single task:

$$T_k = [p_k, time_s^k, time_e^k, x_k, T_A^k, T_{SDC}^k, T_D^k, T_{RD}^k] \quad (1)$$

p_k represents the priority of task k ; $time_s^k$ and $time_e^k$ represent the valid start and end execution times of task k , respectively; x_k indicates whether task k is executed; T_A^k , T_{SDC}^k , T_D^k , and T_{RD}^k correspond to the parameters of the antenna, SDC, demodulator, and RD required for task k , respectively.

3.2. Optimization Objective

The objective of researching the hardware resource scheduling problem in satellite data transmission ground stations is to maximize the reception of critical satellite data transmission with limited resources. Therefore, the following optimization goals are primarily designed to address this issue.

3.2.1. Maximizing Task Execution Rate

If task k can be successfully scheduled to equipment that meets the constraint conditions, then $T_k \cdot x_k = 1$; otherwise, $T_k \cdot x_k = 0$. The formula for calculating the task execution rate (TER) is as follows:

$$TER = \frac{\sum_{1 \leq k \leq n} T_k \cdot x_k}{n} \quad (2)$$

$\{k | 1 \leq k \leq n, n \in Z\}$ represents the range of task id, n denotes the total number of tasks, and x_k indicates whether task k is executed, taking values of 0 or 1.

3.2.2. Maximizing Task Revenue

For a given task list T , where the revenue-influencing factors for each task are defined as the effective execution time and task priority, the formula for calculating the total task revenue (TTR) of task list T is as follows:

$$TTR = \sum_{1 \leq k \leq n} x_k \cdot p_k \cdot (time_e^k - time_s^k) \quad (3)$$

$\{k | 1 \leq k \leq n, n \in Z\}$ represents the range of task id, n denotes the total number of tasks, and $time_e^k - time_s^k$ indicates the effective execution time duration of task k .

3.3. Constraint Condition

3.3.1. Importance of Data Transmission Tasks

During the hardware resource scheduling process at the ground station, priority should be given to the execution of data transmission tasks with a priority level of 2 or 3. That is, ground station hardware resources should be allocated in descending order of data transmission task priority, where the data transmission task priority levels include level 1, level 2, and level 3.

3.3.2. Ground Resource Constraints

Considering the ground resource constraints, we focus on four types of equipment resources: antennas, SDCs, demodulators, and RDs. The involved constraints include those related to the resource occupation time of data transmission tasks and the capabilities of each resource. We describe these constraints in detail below.

If a visible time window meets the user's data transmission requirements and is converted into task k , then the effective execution time of task k must be encompassed within it. The specific expression is shown in Equation (4).

$$\begin{aligned} time_{WL_j}^s < time_s^k < time_e^k < time_{WL_j}^e, \\ \forall j \in WL.Len, \forall k \in [1, n] \end{aligned} \quad (4)$$

$time_{WL_j}^s$ represents the start time of the visible time window j between the satellite and the ground station; $time_{WL_j}^e$ represents the end time of the visible time window j between the satellite and the ground station; and $WL.Len$ denotes the number of visible windows.

All the parameters of DL_i^k assigned to task k must match the required demodulator parameters for task k . The specific expression is shown in Equation (5).

$$\begin{cases} T_D^k.demMode \in DL_i^k.demMode, \forall k \in [1, n] \\ T_D^k.codeRate \in DL_i^k.codeRate, \forall k \in [1, n] \\ T_D^k.decMode \in DL_i^k.decMode, \forall k \in [1, n] \\ i \in [1, DL.Len] \end{cases} \quad (5)$$

$demMode$ indicates the demodulation method; $codeRate$ represents the code rate; $decMode$ represents the decoding method; and $DL.Len$ denotes the number of demodulator resources at the ground station.

The parameters of AL_i^k assigned to task k must be able to match the required antenna parameters for task k . The specific expression is shown in Equation (6).

$$\begin{cases} T_A^k.polar \in AL_i^k.polar, \forall k \in [1, n] \\ T_A^k.bands \in AL_i^k.bands, \forall k \in [1, n] \\ i \in [1, AL.Len] \end{cases} \quad (6)$$

where $polar$ indicates the antenna polarization mode; $bands$ represent the antenna operating frequency range; and $AL.Len$ denotes the number of antenna resources at the ground station.

At any given time, each antenna, SDC, and demodulator can only be allocated to one task for use. Additionally, due to the potential for data confusion and loss when multiple data transmission tasks share the same RD to perform their respective tasks, the use of a shared RD is not considered when there are sufficient resources available.

The operating input frequency and output frequency of the SDC assigned to task k must support the input frequency and output frequency required for task k . The specific expression is shown in Equation (7).

$$\begin{cases} T_{SDC}^k.inputFreq \leq SDCL_i^k.inputFreq, \forall k \in [1, n] \\ T_{SDC}^k.outFreq \leq SDCL_i^k.outFreq, \forall k \in [1, n] \\ i \in [1, SDCL.Len] \end{cases} \quad (7)$$

$inputFreq$ represents the input frequency; $outFreq$ represents the output frequency; and $SDCL.Len$ denotes the number of SDCs at the ground station.

The parameters of the RDL_i^k assigned to task k must match the required RD parameters for the task. The specific expression is shown in Equation (8).

$$\begin{cases} T_R^k.codeRate \leq RDL_i^k.codeRate, \forall k \in [1, n] \\ T_R^k.inputMode \in RDL_i^k.inputMode, \forall k \in [1, n] \\ i \in [1, RDL.Len] \end{cases} \quad (8)$$

$codeRate$ represents the code rate; $inputMode$ indicates the input method, with TCP/IP protocol being a common input mode for RD; and $RDL.Len$ denotes the number of RDs at the ground station.

The number of crossings through the IFSM occupied by successfully allocated tasks must not exceed N_{cm} . The specific expression is shown in Equation (9).

$$\sum_{1 \leq k \leq n} x_k x_c^k < N_{cm} \quad (9)$$

x_c^k represents whether the equipment assigned to task k crosses IFSM; N_{cm} denotes the maximum allowed number of crossings through the IFSM supported by the ground receiving system.

4. ITGA Design

The GA is an intelligent search algorithm with excellent performance [17]. However, when dealing with large-scale problems, due to its inherent disadvantages, such as slow convergence speed and susceptibility to getting trapped in locally optimal solutions [18], the GA often struggles to ensure efficient solution finding. Therefore, we propose combining the TSA with the crossover and mutation operators in the GA to form improved crossover and mutation algorithms that can enhance the efficiency of the GA in finding solutions. The execution flow of the improved tabu genetic hybrid algorithm is illustrated in Figure 4.

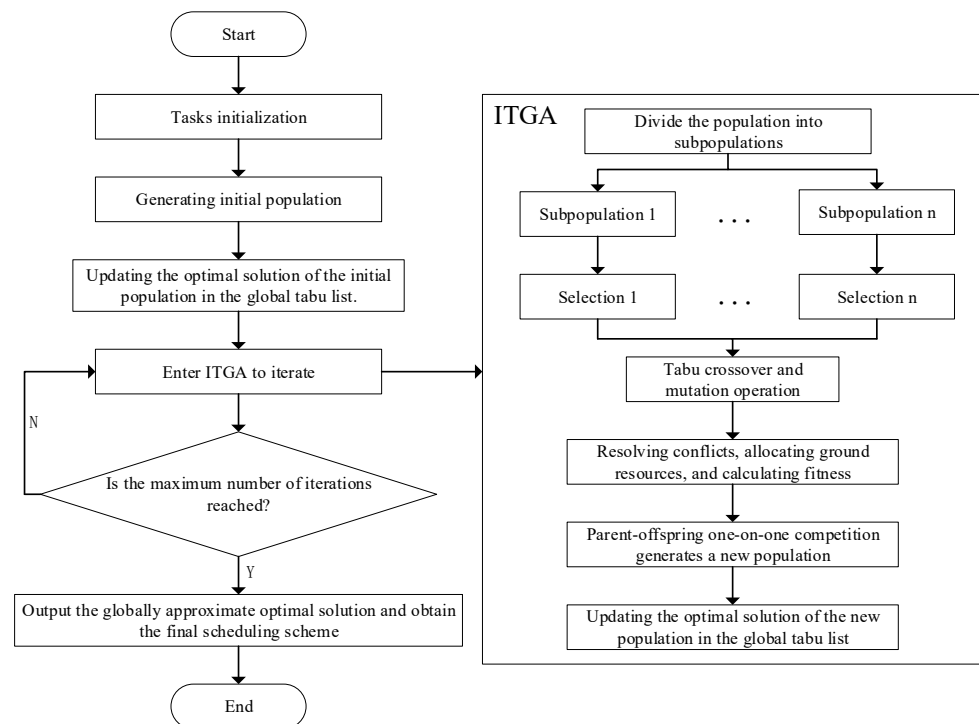


Figure 4. Flow chart of ITGA.

After initializing the data transmission tasks to form a task set, the algorithm employs a task set partitioning method to divide the set into multiple sub-task sets based on data transmission conflicts. Each sub-task set is then solved separately. Within the same sub-task set, data transmission tasks may have conflicts, while tasks in different sub-task sets do not conflict with each other. Utilizing the task set partitioning approach can reduce the traversal of non-conflicting data transmission tasks to some extent, thereby decreasing algorithm runtime and enhancing computational efficiency.

Furthermore, during the process of solving the task set, it is necessary to prune the data transmission durations of tasks reasonably in response to conflicts within the set, aiming to allocate ground station hardware resources to each data transmission task as much as possible. The following heuristic rules are defined as principles for the conflict resolution algorithm:

1. Only prune data transmission tasks with a priority of 1, ensuring effective data transmission durations for tasks with priorities of 2 or 3.
2. The remaining effective data transmission duration after task pruning should not be less than the user's satellite data transmission requirements. If the effective duration is insufficient, the task is abandoned.
3. Select the solution with the shortest pruned effective data transmission time and the longest actual effective data transmission time.
4. All data transmission tasks that are not covered by the effective data transmission times of other tasks should be ensured to allocate receiving resources for data transmission.

4.1. Encoding

The algorithm employs a non-repeating integer encoding scheme to address the hardware resource scheduling problem for satellite data transmission ground stations. Compared to other encoding methods, this approach generates chromosomes that more intuitively represent the hardware resource allocation sequence for satellite data transmission tasks. The chromosome structure is illustrated in Figure 5.

5	1	2	3	4	6
---	---	---	---	---	---

Figure 5. Chromosome: the number indicates the task number, and the location sequence indicates the execution sequence.

The integer values within the chromosome individual correspond to the task id, denoted as k in Equation (1). Data transmission tasks that are positioned earlier in the sequence receive priority in the allocation of ground station hardware resources based on their demand and the constraints imposed by ground resources. If the required hardware resources for data transmission tasks positioned later in the sequence are insufficient, those tasks may be abandoned.

4.2. Fitness Evaluation

Fitness value is a crucial metric that determines the quality of each chromosome. Equation (10) is selected as the fitness function for the ITGA. This fitness function represents both the data volume and the value of data received by the ground station from satellite transmission tasks, aligning with the objective of hardware resource allocation at the ground station.

$$f(x) = \sum_{1 \leq k \leq n} x_k \cdot p_k \cdot (time_e^k - time_s^k) \quad (10)$$

4.3. Generating Initial Population

Firstly, the initial population size for the ITGA is set to n . Secondly, for tasks with conflicting data transmission times, a heuristic rule-based conflict resolution method is applied to adjust the transmission schedules, ensuring they become executable for data transmission. Subsequently, several chromosomes are randomly generated, and heuristic rules are used to allocate ground station hardware resources, aiming to improve the fitness values of these chromosomes. Finally, based on their fitness values, the top n optimal chromosomes are selected to form the initial population.

The heuristic rule-based ground station hardware resource allocation method is as follows:

1. Firstly, based on the equipment parameters required for the task, filter out equipment that does not meet the constraint conditions according to Formulas (4)–(9). Secondly, allocate the remaining available equipment to the current task. Finally, repeat the above operations until all task equipment is allocated.
2. During the allocation process, if there is a conflict in equipment usage and no equipment of the same type is available, the following allocation strategy applies: if the priority task that has already been allocated equipment has a longer effective data

transmission duration than subsequent tasks, it will not be preempted by those subsequent tasks with conflicting equipment requirements.

3. If a data transmission task does not have any time conflicts with other data transmission tasks, it will be directly allocated the most optimal equipment.
4. When multiple devices of the same type on the same side of the switch matrix are capable of serving the same task and are not occupied by other tasks, the device with the fewest allocated tasks will be selected to serve that task, and the allocation plans for the other devices will not be considered.

The flow chart of the initial population generation is shown in Figure 6.

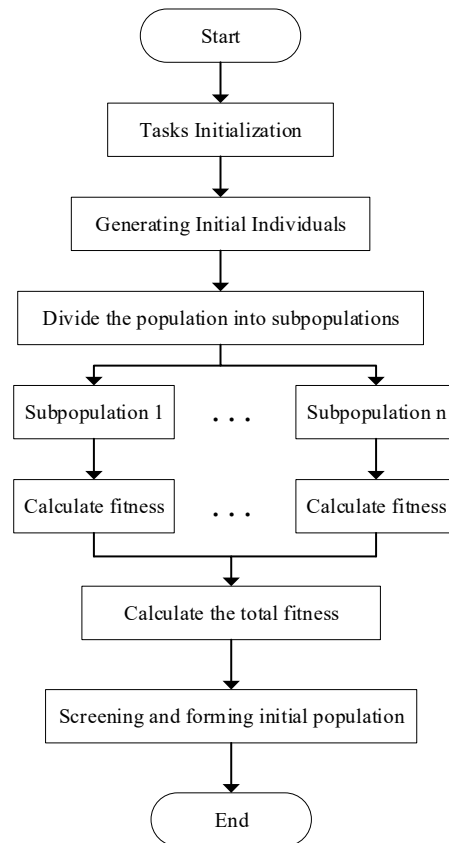


Figure 6. Flow chart of the initial population generation.

4.4. Selection Operator

We adopt the tournament selection (TS) method for selection operations. The calculation process of the TS method is shown in Algorithm 1.

Algorithm 1: TS Method

Input: Pop, n and $fits$ ›The population to be selected, the number of individuals selected each time, and the corresponding fitness value list of the population to be selected.

Output: Pop_{new} ›The new population generated by Pops.

```

1: while  $Pop.Len > Pop_{new}.Len$  do ›Len indicates the size of the population.
2:    $Pop_n \leftarrow$  Select  $n$  individuals from  $Pop$ 
3:    $fit_n \leftarrow$  Select  $n$  fitness value from  $fits$  that correspond one-to-one with  $Pop_n$ 
4:   Sort  $Pop_n$  in descending order based on the values in  $fit_n$ 
5:   Add the  $Pop_n$  with the highest  $fit_n$  value to  $Pop_{new}$ 
6: end while
7: return  $Pop_{new}$ 
  
```

After applying the selection operator, a new population is generated. This new population competes for survival with the initial population using a one-on-one survivor competition (OSC) Method (Algorithm 2). Specifically, each chromosome from the two populations under consideration is evaluated and the fitter ones are selected. Finally, the resulting population is output.

Algorithm 2: OSC Method

Input: Pop_{parent} , Pop_{child} , $fits_{child}$ and $fits_{parent}$ › Parent population waiting for OSC, Offspring population waiting for OSC, Fitness value list corresponding to the offspring population, and Fitness value list corresponding to the Parent population.

Output: Pop_{parent} and $fits_{parent}$ › The new parent population generated by OSC and the new parent fits generated by OSC.

```

1: for  $i = 1: Pop_{parent}.Len$  do › Len indicates the size of the population.
2:   if  $fits_{parent}[i] < fits_{child}[i]$  then
3:      $fits_{parent}[i] = fits_{child}[i]$ 
4:      $Pop_{parent}[i] = Pop_{child}[i]$ 
5:   end if
6: end for
7: return  $Pop_{parent}$ 
  
```

4.5. Crossover and Mutation Operations

The process of selecting two populations as parent populations using the TS method and performing crossover operations with crossover probability r_c is as follows: Firstly, two parent chromosomes are selected from each of the two parent populations. Secondly, two target points are randomly chosen, and the sequences between these two target points in the two parent chromosomes are swapped. Through a mapping approach, duplicate chromosomal genes are avoided, ensuring that the newly generated offspring chromosomes also adhere to the encoding rules. Finally, the two generated offspring chromosomes are added to the offspring population. This crossover operation is repeated for the parent populations until the offspring population is fully generated. The crossover operation process is illustrated in Figure 7.

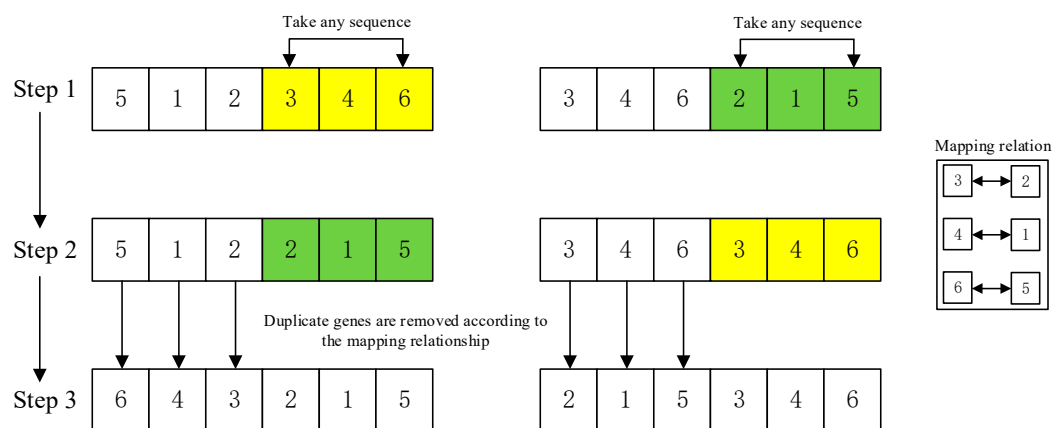


Figure 7. Crossover operation. Different colors indicate chromosomal position changes.

In addition, due to the potential for excessively long search times required for unrestricted exchange sequence lengths, it is stipulated that the length of the exchange sequence must be less than 10% of the chromosome length.

After obtaining the offspring population through crossover operations, mutation operations are performed on the offspring population according to the mutation probability. In this paper, the swap operator is used for chromosome mutation. The operation steps are as follows: two gene positions, P1 and P2, are randomly selected on the chosen chromosome,

and the genes corresponding to these positions are swapped to obtain a new chromosome. The operation process is shown in Figure 8.

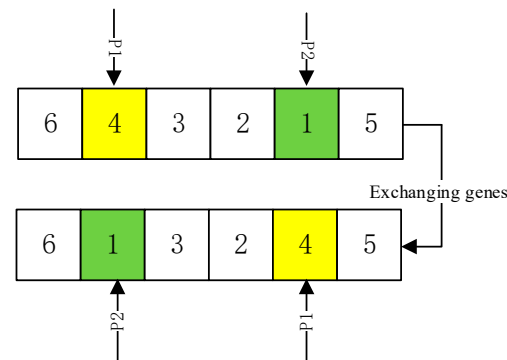


Figure 8. Mutation operation. P1, P2, and different colors indicate chromosomal position changes.

4.6. TSA

The TSA can avoid repeatedly visiting already searched chromosomes by introducing a tabu list during the local search process, thus helping the GA avoid getting trapped in local optimal solutions [19].

In this paper, the TSA is used to improve the crossover and mutation operators of the GA: a local tabu list is established in the crossover operator, and this list, combined with a global tabu list that stores the optimal solutions generated in each iteration, can avoid repeated searches by the crossover and mutation operators. The execution process of the tabu crossover (TC) operator is shown in Algorithm 3.

Algorithm 3: TC operator

Input: Pop₁ and Pop₂ › population 1 waiting for TC operation and population 2 waiting for TC operation.

Output: Pop_{crossover} › The new population generated by TC operation.

1: Initialize Flag = True

2: **for** $i = 1: \text{Pop}_1.\text{Len}$ **do** › Len indicates the size of the population.

3: Flag ← True

4: **while** Flag != False **do**

5: $g_1, g_2 \leftarrow \text{Crossover}(\text{Pop}_1, \text{Pop}_2)$ › Pop₁ and Pop₂ generate new chromosomes g_1 and g_2 through the crossover operation described in Section 4.5.

6: **if** g_1, g_2 do not exist in the local and global tabu lists, as well as in Pop_{crossover} **then**

7: Pop_{crossover} ← g_1, g_2

8: Flag ← False

9: **end if**

10: **end while**

11: **end for**

12: **return** Pop_{crossover}

Furthermore, the concept of a neighborhood solution set is introduced into the mutation operator to enhance the local search capability of the genetic algorithm. The definitions related to the tabu mutation (TM) operator are as follows:

1. Initial solution (IS): represents the chromosome individual initially involved in mutation.
2. Neighborhood solution (NS) set: a collection of solutions generated in some manner based on the initial solution.
3. Candidate solution (CS): the optimal solution with the highest fitness value within the neighborhood solution set is compared with the initial solution to determine whether it should replace the initial solution.

The execution process of the TM operator is outlined in Algorithm 4.

Algorithm 4: TM operator

Input: Pop › The population waiting for TM operation
Output: Pop_{mutate} › The new population generated by TM operation.

```

1: Initialize Flag = True
2: for  $i = 1$ : Pop.Len do › Len indicates the size of the population.
3:   Flag ← True
4:   while Flag != False do
5:      $g_1 \leftarrow \text{Mutate}(\text{Pop}[i])$  › IS generate new chromosomes  $g_1$  through the mutate operation
described in Section 4.5.
6:     PopNS ← Mutate (Pop[i]) ›Applying mutation operations to PopNS produces several
neighborhood solutions, thereby forming a neighborhood solution set.
7:      $g_2 \leftarrow \text{Pop}_{\text{NS}}$  ›Finding the  $g_2$  (CS) from NS
8:     if  $g_2$  is superior to  $g_1$  then
9:       best ←  $g_2$ 
10:    else
11:      best ←  $g_1$ 
12:    end if
13:    if best do not exist in the local and global tabu lists, as well as in Popmutate then
14:      Popmutate ← best
15:      Flag ← False
16:    end if
17:  end while
18: end for
19: return Popcrossover

```

4.7. Termination Condition

Set the maximum number of iterations to Generations. When the algorithm reaches the specified number of Generations, the optimal chromosome solution from the latest population is selected as the final ground station resource allocation result.

5. Experiment Results and Analysis

To validate the effectiveness of the ITGA, this paper conducts simulation experiments on the algorithm. The algorithm is programmed in Python language, specifically version 3.10, and solved on a computer with an AMD Ryzen 7 processor running at 3.2 GHz, using 16 GB of RAM and the Windows 10 operating system.

The algorithm parameters are set as follows: population size Pop = 50, maximum number of iteration Generations = 150, crossover probability = 0.9, mutation probability = 0.1, and tabu list length = 35.

The satellite Two-Line Element (TLE) [20] data are sourced from SAIC's Space-Track, a global satellite and rocket data network. For the purpose of this experiment, a single remote sensing satellite ground station located at latitude 40.5°, longitude 116.9°, and an altitude of 0.057 km is selected. Furthermore, four experimental scenarios are designed for mission planning.

In terms of satellite mission scheduling, this paper utilizes the STK [21] tool to calculate visible time windows and assigns a task to each visible window exceeding 6 min. Table 2 lists the number of tasks, total fitness, and the total mission duration for different experimental scenarios, identified by unique numbers with specific details concealed. Table 3 presents the ground station resources, including antennas, SDCs, demodulators, and RDs.

Table 2. Experimental scene scale.

Case	Number of Tasks	Total Duration	Total Fitness
1	41	21,077	46,159
2	64	24,008	49,611
3	67	20,987	32,202
4	100	47,214	89,566

Table 3. Ground station resources.

Type	Number	Parameter
7.5 m Antenna	3	Operating mode: dual circular polarization frequency reuse; operating frequency bands: L, X.
L SDC	2	Primarily utilized for signal frequency conversion in the L-band.
X SDC	2	Primarily utilized for signal frequency conversion in the X-band.
X Demodulator	2	Primarily utilized for demodulation tasks in X-band links; decoding methods include RS code, CRC decoding, differential decoding, convolutional code, and forward error correction; differential decoding: NRZ_L, NRZ_M, etc.; demodulation schemes: BPSK, QPSK, etc.; demodulation code rate: 10~800 Mbps.
L Demodulator	2	Primarily utilized for demodulation in L-band links; demodulation code rate ranging from 10 to 640 Mbps; other parameters are identical to those of the X Demodulator.
RD	4	Input method: TCP/IP protocol format; maximum single-channel code rate of 600 Mbps with a combined code rate for five channels not exceeding 1200 Mbps.

To analyze the performance of the ITGA, we introduced several comparison algorithms, including a TGA without conflict resolution and task set partitioning methods, a DFS-integrated improved GA (DFSGA), a First-Come-First-Served Algorithm (FCFSA) [22], and a Priority Algorithm (PA). The DFSGA determines the chromosome individuals representing the task execution order through the genetic operations of the GA. It allocates ground station resources using DFS to calculate the fitness value of the chromosome individuals. The best individuals from the initial population are selected for the next iteration until the end of the iteration process. Finally, the optimal allocation scheme is determined by selecting the best individual from the final population. On the other hand, the FCFSA and the PA prioritize tasks based on execution time or task priority, respectively, for ground station resource allocation.

We use average running time (ART), average fitness value (AFV), mean deviation (MD), average number of completed tasks (ANCT), average task execution rate (ATER), average non-ordinary task execution rate (ANOTER), and average convergence generation (ACG) as metrics to evaluate algorithm efficiency, convergence speed, optimization ability, and stability. However, due to the stochastic nature of the GA, the obtained results may not always be globally optimal solutions [23]. Therefore, in this study, we conducted simulation experiments by calculating the average of 10 runs for each experimental scenario. The simulation results are presented in Table 4. Notably, the “0” in the mean deviation metric indicates that the algorithm converged early in the running process. This occurred because the data transmission task set had fewer conflicts, and the algorithm found a globally optimal solution during the initial population generation. The “-” symbol indicates that the PA and the FCFSA only consider a single metric, leading to poor stability and rendering the average difference metric ineffective for them.

The simulation results indicate that the ITGA performs well across various metrics in four different-sized experimental scenarios. Compared to the greedy-based FCFSA and the PA, the ITGA achieves an improvement of approximately 3 to 19 tasks in the ANCT metric, an increase of about 6.61% to 19.8% in the ATER metric, and an enhancement of roughly 10.59% to 21.62% in the ANOTER metric. Since the FCFSA and the PA do not involve an iterative process, no comparison is made for the ART metric.

Table 4. Comparison of the results of the algorithms.

Case	Algorithm	MD	ANCT	ART	AFV	ATER	ANOTER
1	DFSGA	0	38	19.71	44,029	92.68%	94.12%
	TGA	0	38	12.85	44,029	92.68%	94.12%
	ITGA	0	38	12.35	44,029	92.68%	94.12%
	FCFSA	-	34.7	4.36 (ms)	39,007.7	84.63%	84.71%
	PA	-	34.5	4.51 (ms)	37,738.3	84.15%	83.53%
2	DFSGA	183.4	45	24.35	36,636	70.31%	66.09%
	TGA	131.2	45	14.36	37,399.4	70.31%	77.27%
	ITGA	125.8	46	12.95	37,720	71.88%	77.27%
	FCFSA	-	36.6	6.12 (ms)	30,683.9	57.19%	64.55%
	PA	-	37.8	6.05 (ms)	29,539.4	59.06%	63.86%
3	DFSGA	0	54	28.75	31,768	96.43%	100.00%
	TGA	0	54	16.71	31,768	96.43%	100.00%
	ITGA	0	54	15.02	31,768	96.43%	100.00%
	FCFSA	-	50.3	7.17 (ms)	29,712.2	89.82%	94.39%
	PA	-	50.6	7.79 (ms)	29,604.8	90.36%	91.46%
4	DFSGA	127.5	93.7	52.24	85,544.9	93.40%	97.14%
	TGA	55.3	95	25.19	86,750	95.00%	98.39%
	ITGA	9.5	95	23.78	86,750	95.00%	98.39%
	FCFSA	-	77.2	13.11 (ms)	73,876.3	77.20%	86.94%
	PA	-	75.2	14.42 (ms)	68,601.2	75.20%	76.77%

When compared to the TGA, the ITGA employs a heuristic rule-based conflict resolution approach. This leads to an improvement of one task execution in the ANCT metric and a 1.57% increase in the ATER metric specifically in experimental scenario 2, which has a higher number of data transmission conflicts. However, in scenarios 1, 3, and 4, where conflicts are fewer or cannot be resolved by trimming transmission durations, both algorithms perform similarly. This underscores the effectiveness of the conflict resolution method in scenarios with more conflicts. Regarding the ART metric, ITGA achieves a speedup of 0.5 s to 1.69 s due to the introduction of the task set partitioning method. This reduction in the traversal of conflict-free tasks, combined with the parallelism of the GA and asynchronous multitasking execution techniques, allows for simultaneous solution finding for sub-task sets, ultimately enhancing algorithm efficiency. In comparison to the DFSGA, the ITGA demonstrates improvements in the ANCT, ATER, and ANOTER metrics by one task, 1.57%, and 11.18% in scenarios 2 and 4, respectively. In scenarios 1 and 3, both algorithms perform similarly due to fewer conflicts in the data transmission task set. As for the ART metric, ITGA achieves an improvement of 7.36 s to 28.46 s.

In addition, due to the existence of numerous conflicting time periods within the satellite-to-ground visibility windows in experimental case 2, the scheduling outcomes achieved in this particular scenario are better suited to demonstrate the effectiveness of the ITGA. Therefore, this study carries out a detailed analysis of four key metrics in experimental case 2: ground resource allocation results, distribution of ground station resource usage time, algorithm convergence speed, and algorithm stability.

The task scheduling outcomes achieved by the ITGA in experimental case 2 are illustrated in Figure 9.

The figure illustrates the task schedule for satellites over a 24 h period, with the *x*-axis representing time of day in seconds and the *y*-axis denoting satellite names. The small squares in the graph represent the individual tasks assigned to each satellite throughout the day. Red squares indicate tasks that were not executed due to an unsuccessful allocation of ground station hardware resources, while green squares represent tasks that were successfully allocated resources and executed.

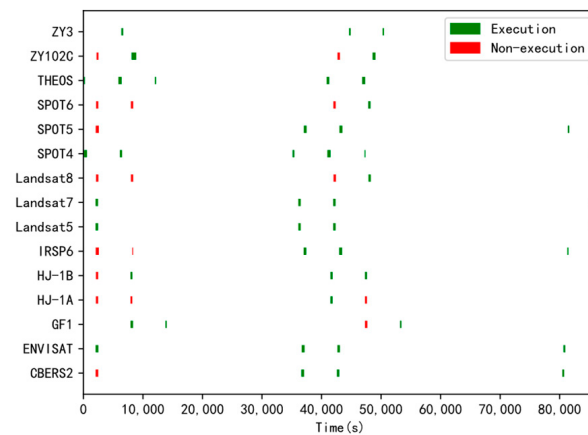


Figure 9. Ground resource allocation results solved by the ITGA in case 2.

In the scheduling results, there were 17 data transmission tasks that could not be allocated ground station hardware resources and were therefore scheduled not to be executed. Among these, 10 were important or urgent tasks that could not be allocated resources due to their data transmission durations being shorter than those of other critical tasks. One task could not be allocated ground station equipment due to a link conflict and was therefore abandoned. The remaining six tasks were deemed ordinary priority and could not be accommodated through trimming their transmission durations, leading to their abandonment.

The distribution of antenna usage time is shown in Figure 10.

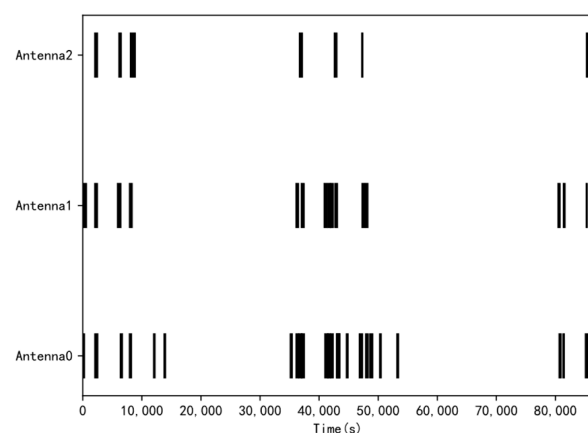


Figure 10. ITGA antenna usage time distribution diagram.

The black squares represent moments when a particular antenna is being utilized for a satellite data transmission task, with the width of the square indicating the duration of antenna occupancy. Due to our consideration of load balancing during equipment allocation, the antenna usage time distribution is relatively uniform. Within any conflicting time period, there is no situation where satellite data transmission tasks are concentrated on a single antenna for execution. If an antenna is already occupied by a transmission task, other transmission tasks will be assigned to idle antennas. During non-conflicting periods, load balancing is not considered due to the absence of equipment quantity bottlenecks. Additionally, since other equipment exhibits similar effects, further elaboration is not necessary.

Regarding the ACG metric, the FCFSA and PA algorithms based on greedy strategies do not involve an iterative process. Therefore, in the comparison of ACG metrics, these two algorithms are not considered. The comparison of ACG metrics for the remaining three algorithms is presented in Table 5.

Table 5. Case 2 comparison of ACG of each algorithm.

Case	Algorithm	ACG
2	DFSGA	70.9
	TabuGA	52.6
	ITabuGA	23.9

As shown in the ACG metric comparison chart, the GA enhanced with DFS partially addressed the issue of slow convergence due to its weak local search capability. However, its convergence speed is still slow compared to the TGA and the ITGA. Both the TGA and the ITGA achieve convergence within 60 generations, with the ITGA exhibiting a 54.56% reduction in average convergence generations. This demonstrates that the ITGA can converge more rapidly, as illustrated in Figure 11.

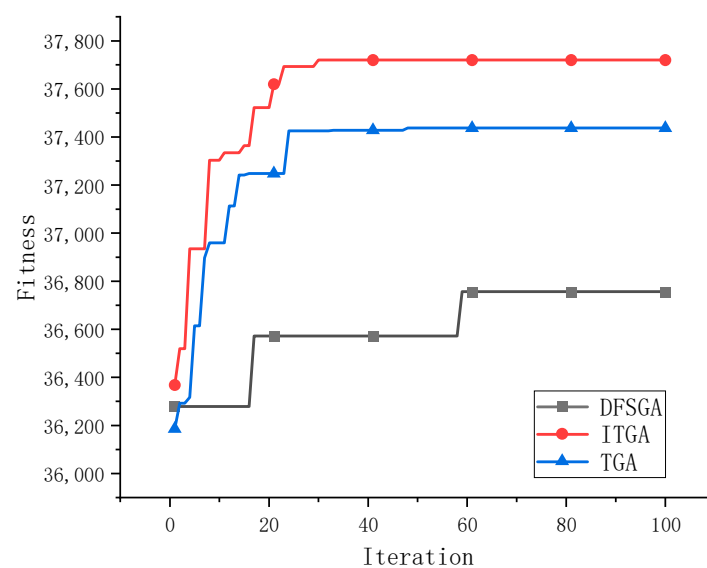
**Figure 11.** Comparison of convergence speeds of various algorithms in a specific experiment.

Figure 11 shows the comparison of algorithm convergence for one of the ten experiments conducted in experimental scenario 2. Due to space limitations, the performance of the remaining experiments is not shown, but they exhibit similar trends to Figure 11. As shown in Figure 11, the TGA can continuously escape local optimal solutions in the early stages of the algorithm and iteratively evolves towards the global approximate optimal solution. The task-set-partitioning- and conflict-resolution-optimized TGA (ITGA) further enhances local search capabilities while improving task completion rates by pruning the data transmission time of conflicting tasks, ultimately increasing the fitness value of the solution. While the DFSGA can also escape local optimal solutions, it requires a certain number of iterations to do so. In all ten experiments, the ITGA outperforms both the TGA and the DFSGA in terms of fitness and iteration, demonstrating the effectiveness of the ITabuGA approach.

Additionally, as illustrated in Figure 12, the ITGA algorithm was able to identify the ground station hardware resource scheduling scheme with the maximum total fitness value in all ten experiments of Case 2. It also performed better than the DFSGA and the TGA in terms of the ANCT metric. Both the FCFSA and the PA exhibited unsatisfactory stability performance. In summary, the ITGA's stability performance is superior to that of other algorithms.

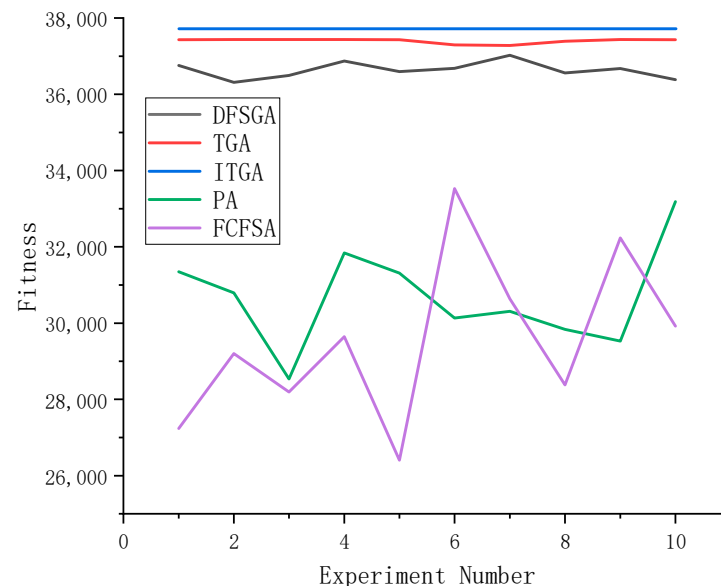


Figure 12. Comparison of algorithm stability.

6. Conclusions

The hardware resource scheduling problem for satellite data transmission ground stations is a multi-constraint optimization problem characterized by limited resources, complex constraints, and significant difficulties standing in the way of solutions. In this paper, based on the actual hardware resource situation of satellite ground stations, a constraint satisfaction model for satellite data transmission tasks is established by analyzing the optimization objectives and constraints of the ground station hardware resource scheduling problem. During the solution process, the set of pending data transmission tasks is first divided into several subsets according to certain principles, and each subset is solved separately. The results are then aggregated to form an initial solution, which is further screened to form an initial population for subsequent algorithm iterations. By comparison with benchmark algorithms, the effectiveness and feasibility of the hybrid GA in solving the hardware resource scheduling problem for satellite data transmission ground stations are demonstrated. The experimental results show that the improved algorithm proposed for the first time in this paper not only enhances the convergence speed and stability of the TGA but also improves the efficiency of hardware resource utilization in ground stations, effectively addressing the hardware resource scheduling problem for satellite data transmission ground stations.

Due to the limitations of constraints, this paper only considers conventional scheduling scenarios. Future research could explore the handling of emergency situations, such as ground resource failures and temporary urgent tasks. In the event of such emergencies, the system could automatically replan and redistribute hardware resources at ground stations to further validate and strengthen our research.

Author Contributions: Conceptualization, C.Y. and M.G.; methodology, L.X.; software, L.X. and B.W.; validation, L.X., M.G. and C.Y.; data curation, L.X.; writing—original draft preparation, L.X.; writing—review and editing, L.X. and C.Y.; supervision, L.X. and C.Y.; funding acquisition, C.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by the National Natural Science of Foundation of China (Grant No. 61871468), the Natural Science of Foundation of Zhejiang Province (Grant No. LY22F010013), and the Natural Science of Foundation of NingBo city (Grant No. 2022J067).

Data Availability Statement: Data is contained within the article.

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

References

1. Fan, H.J.; Zhang, W.Y.; Tian, M.M.; Ma, G.B.; Cheng, B. A resource scheduling method for satellite mission ground station based on particle swarm optimization algorithm. *J. Univ. Chin. Acad. Sci.* **2022**, *39*, 801.
2. Chen, H.; Sun, G.; Peng, S.; Wu, J.J. Dynamic rescheduling method for measurement and control data transmission resources based on multi-objective optimization. *J. Syst. Eng. Electron.* **2023**, 1–12. Available online: <http://kns.cnki.net/kcms/detail/11.2422.TN.20230220.1046.006.html> (accessed on 16 October 2022).
3. Küçük, M.; Yıldız, Ş.T. A constraint programming approach for agile Earth observation satellite scheduling problem. In Proceedings of the 2019 9th International Conference on Recent Advances in Space Technologies (RAST), Istanbul, Turkey, 11–14 June 2019; pp. 613–617.
4. Sun, W.J.; Ma, G.B.; Tian, M.M.; Lin, Y.M.; Huang, P. Remote sensing satellite ground station antenna intelligent scheduling with LSTM and heuristic search. *J. Univ. Chin. Acad. Sci.* **2022**, *39*, 532–542.
5. Huang, S.L.; Ma, D.Q.; Fang, D.M.; Cui, T. Satellite data transmission scheduling based on improved ant colony system. *Radio Eng.* **2015**, *45*, 27–30+58.
6. Liu, J.Y.; Tian, M.M.; Huang, P.; Lin, Y.M.; Ma, G.B. Hybrid Decomposition Algorithm for Remote Sensing Satellite Ground Station Resource Scheduling. *J. Jilin Univ.* **2020**, *58*, 611–619.
7. Li, Y.; Yuan, H.Y.; Yu, J.Q.; Zhang, G.W.; Liu, K.P. A Review of the Application of Genetic Algorithms in Optimization Problems. *J. Shandong Ind. Technol.* **2019**, *12*, 242–243+180.
8. Zhang, M.; Hu, X.X. Scheduling of multi-satellite based on genetic algorithm. *J. Hefei Univ. Technol.* **2017**, *40*, 995–999+1008.
9. Antoniou, M.; Petelin, G.; Papa, G. On formulating the ground scheduling problem as a multi-objective bilevel problem. In Proceedings of the International Conference on Bioinspired Methods and Their Applications, Brussels, Belgium, 19–20 November 2020; pp. 177–188.
10. Xhafa, F.; Ip, A.W. Optimisation problems and resolution methods in satellite scheduling and space-craft operation: A survey. *Enterp. Inf. Syst.* **2021**, *15*, 1022–1045. [CrossRef]
11. Chen, H.; Zhai, B.R.; Wu, J.J.; Du, C.; Li, J. A satellite observation data transmission scheduling algorithm oriented to data topics. *Int. J. Aerosp. Eng.* **2020**, *2020*, 2180674. [CrossRef]
12. Zhou, Y.R.; Chen, H.; Li, L.M.; Chen, L.; Jing, N. Immune genetic algorithm for satellite data transmission scheduling. *J. Chin. Comput. Syst.* **2015**, *36*, 2725–2729.
13. Gao, X.Z.; Guo, Y.N.; Ma, G.F.; Zhang, H.B.; Li, W.B. Agile satellite autonomous observation mission planning using hybrid genetic algorithm. *J. Harbin Inst. Technol.* **2021**, *53*, 1–9.
14. Zhang, P.; Feng, X.X.; Ge, X.Q. A hardware resource allocation method for multi-antenna ground station based on improved genetic algorithm. *Comput. Eng. Sci.* **2017**, *39*, 1155–1163.
15. Jiang, T.; Li, H. Sensing Satellite Research on Computer Control Technology for Receiving System of Remote. *J. Comput. Meas. Control* **2013**, *21*, 2434–2435.
16. Zhang, W.L. *Research on Satellite Ground Station Resource Scheduling Optimization Based on Improved Genetic Algorithm*; National Marine Environmental Forecasting Center: Beijing, China, 2013.
17. D'Angelo, G.; Palmieri, F. GGA: A modified genetic algorithm with gradient-based local search for solving constrained optimization problems. *Inf. Sci.* **2021**, *547*, 136–162. [CrossRef]
18. Katoch, S.; Chauhan, S.S.; Kumar, V. A review on genetic algorithm: Past, present, and future. *Multimed. Tools Appl.* **2021**, *80*, 8091–8126. [CrossRef] [PubMed]
19. Umam, M.S.; Mustafid, M.; Suryono, S. A hybrid genetic algorithm and tabu search for minimizing makespan in flow shop scheduling problem. *J. King Saud Univ. Comput. Inf. Sci.* **2022**, *34*, 7459–7467. [CrossRef]
20. Bizalion, H.; Guillot, A.; Petit, A.; Lucken, R. Systematic TLE data improvement by neural network for most cataloged resident space objects. *Adv. Space Res.* **2023**, *72*, 2649–2659. [CrossRef]
21. Analytical Graphics Inc.: Software for Digital Mission Engineering. Available online: <https://www.agi.com/> (accessed on 16 October 2022).
22. Tai, L. Comparison of First Come First Service and Short Job First Scheduling Algorithms. *J. Inf. Comput.* **2019**, 79–80+84.
23. Xia, L.; Yu, N.H. Genetic Algorithm-based Small Satellites Range Scheduling. *J. Commun. Technol.* **2013**, *46*, 64–68.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.