

Research Article

An Adaptive Large Neighborhood Search Algorithm for the Satellite Data Transmission Scheduling Problem

Yingguo Chen,¹ Ming Chen,¹ Jun Wen ,¹ Yingwu Chen,¹ and Wen Xiang²

¹College of Systems Engineering, National University of Defense Technology, Changsha 410073, China

²Southwest Electronics and Telecommunication Technology Research Institute, Chengdu 610041, Sichuan, China

Correspondence should be addressed to Jun Wen; jun_wen@aliyun.com

Received 10 September 2019; Revised 23 December 2019; Accepted 15 June 2020; Published 16 September 2020

Academic Editor: Jeremy Straub

Copyright © 2020 Yingguo Chen et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Satellite establishes a satellite ground station to satellite link with a ground station to complete data transmission. However, in China, satellite ground stations only exist within the country. With the rapid increase in the number of satellites, effectively dispatching satellite communications and maximizing the performance of ground stations are necessary. We propose a heuristic adaptive large neighborhood search algorithm (H-ALNS) to solve the satellite data transmission scheduling (SDTS) problem. The algorithm includes two heuristic rules for generating the initial scheme and the conflict reduction process after updating the neighborhood. A heuristic task assignment method is used to select the execution time window and position for the task sequence. The adaptive operator is used to update the delete operator and insert operator weights into the H-ALNS. The quality of the generated planning scheme improved through continuous neighborhood destruction and repair. Through experimental analysis, the algorithm we propose is feasible for solving SDTS problems and surpasses the planning results acquired using other comparison algorithms. The H-ALNS has good prospects for practical engineering applications.

1. Introduction

A satellite is a spatial platform that operates periodically in planetary orbits and performs various tasks. Satellites can be divided into three types: remote sensing satellites, communication satellites, and navigation satellites. After a satellite performs its distinct mission, the mission completion results or acquired data are transmitted to satellite ground stations via a satellite link when passing above the satellite ground stations. Satellite ground stations perform data analysis and processing and then send the data product to the requesting users. Data transmission is an important part in ensuring that the task results successfully reach the user. China only uses domestic locations for satellite ground. The number of ground stations is not sufficient for the number of satellites. Our goal is to make full use of the available satellite ground stations, allowing for more data transmission with the limited resources.

The satellite data transmission scheduling (SDTS) problem refers to the planning of a data downlink scheme prioritizing the highest possible profit among many satellite data

transmission tasks. Profit is a value that can be obtained assuming successful transfer of a task. Since the orbit of the satellite and the location of the ground station are in a fixed position, the satellite and the satellite ground station are not always visible in real time; only when the two are visible to each other can a satellite ground station to satellite link be established for data transmission. The difficulty of the problem lies in the following: (1) the resources are limited, and the task is far beyond the resource carrying capacity, producing an oversubscription problem; (2) the numerous constraints of the problem make it difficult to find a feasible planning scheme; and (3) the limited capabilities of satellites and satellite ground stations add to the difficulty of solving problems.

The satellite data transmission scheduling problem is part of the satellite mission planning field. In recent years, domestic and foreign scholars have conducted extensive research on satellite mission planning. Zhu et al. [1] analyzed the observation of existing satellites in the rapid completion of the task area during an emergency rescue situation. The main goal was to achieve maximum coverage of the area. In

Fatos et al. [2], a description of the satellite mission planning problem is summarized, and a genetic algorithm is proposed to solve such problems. Kolici et al. [3] used a combination of a local search method and a genetic algorithm to solve the problem of large-scale satellite and ground station scheduling. Chu et al. [4] attempted to build a model and solve the online scheduling of agile imaging satellites using the branch and bound algorithm. The genetic algorithm is applicable to satellite data transmission and satellite ground station monitoring and control scheduling. It sets a specific coding method, including heuristic rules in the algorithm, and demonstrated good performance in improving search speeds and optimization effects in [5–7]. A multisatellite mission planning system for environmental monitoring and disaster prediction is proposed in He et al. [8]. The system has both concentrated and scattered features, and a dynamic segmentation method for polygon regions is constructed. He et al. [9] considered the real-time occurrence of tasks. A new type of online mission planning system for imaging satellites is designed, and an execution mechanism for autonomous mission planning is proposed. Karapetyan et al. [10] proposed a metaheuristic algorithm to achieve the effect of rapid generation of solutions based on the data downlink problem for Canadian satellites. Concerns with prior research of satellite mission planning problems include using their own research to extract scientific problems from practical problems, building models, providing solving algorithms, paying more attention to practical applications, and having a lack of uniform comparison standards.

The adaptive large neighborhood search (ALNS) is a flexible framework for optimizing problems through adaptive neighborhood structure improvements [11, 12]. Neighborhood structure improvement includes two methods: destroy method and repair method. ALNS improves the optimization by adaptively selecting to update neighborhood. ALNS algorithm has been well utilized in the field of planning and scheduling. Ropke and Pisinger [13] designed a large neighborhood search algorithm with heuristic rules for a pickup and delivery problem with time windows, using historical information to improve the optimization process. Demir et al. [14] analyzed the characteristics of the Pollution-Routing Problem (PRP). This problem is an extension of the classical vehicle routing problem. A large neighborhood search algorithm is proposed, and the algorithm is validated through experiments. Hemmelmayr et al. [15] used the adaptive large neighborhood search algorithm to solve the two-echelon vehicle routing problem (2E-VRP) based on the idea of a layered solution and combining the existing operators. Li et al. [16] proposed an effective adaptive large neighborhood search algorithm. This algorithm is compared with the mixed integer programming (MIP) solver, which is excellent in both the quality and the efficiency of the solution.

SDTS is a kind of NP (nondeterministic polynomial) hard problem. It is difficult to solve the large-scale scenario in an effective time using the exact solution algorithm. We propose a heuristic adaptive large neighborhood search algorithm for the satellite data transmission scheduling problem. Two heuristic rules are used for the process of postinteraction conflict elimination after the task of planning the initial plan

and using repair methods. The adaptive large neighborhood search algorithm includes the destroy method and the repair method. For the satellite mission planning problem, four deletion operators are set. In the repair method, three insertion operators are set. The adaptive mechanism is also set in H-ALNS, and the weight of the operator is adjusted according to the performance of the algorithm during the run. The original planning scheme was continuously optimized by the H-ALNS algorithm to find a more satisfactory result.

The structure of this paper is as follows. In the second part of this paper, the satellite data transmission planning problem will be introduced, and a mathematical model will be established. The H-ALNS algorithm will be provided in the third part of this paper. Detailed explanations of the heuristic rules, initial scheme generation, and operators for the algorithm will be given. The conclusions of this paper will be summarized in the last section.

The innovation of this paper is that this paper analyzes the problem of satellite data transmission mission scheduling and constructs a mixed integer model of SDTS problem. In this paper, a H-ALNS algorithm is proposed. Based on the original ALNS algorithm, a heuristic task scheduling method is added, and a heuristic rule adapted to SDTS is proposed. At the same time, an acceptance criterion is also designed in H-ALNS to accept the inferior solution temporarily. By this way, H-ALNS can find a higher quality solution in the end.

The structure of this paper is as follows. In the second part of this paper, the satellite data transmission planning problem will be introduced, and a mathematical model will be established. The H-ALNS algorithm will be given in the third part of this paper. The heuristic rules, initial scheme generation, and operation operators in the algorithm will have detailed explanation. The conclusions of this paper will be given in the last part.

The innovation of this paper lies in the following: using our proposed large neighborhood search algorithm to solve satellite data transmission planning problems, obtaining higher-yield task execution schemes in small-scale and large-scale mission scenarios, and providing theoretical guidance and method support for actual satellite scheduling.

2. Problem Description

Satellite data transmission scheduling can be described as a time window selection problem, which aims at selecting data transmission from several satellites to several ground stations and determining the transmission sequence and the execution times without violating constraints. A time window is defined as a satellite passes over a satellite ground station at a time. The data is acquired by satellites or needs to be transmitted.

2.1. Assumptions. The SDTS problem involves technologies in many fields such as communications, aerospace, and control. It is a complex and comprehensive problem. In order to highlight the characteristics of the planning, a mathematical model is built to solve the problem, and the following assumptions are made for the problem:

- (1) All ground station to satellite links are full duplex, and the established link is stable and reliable
- (2) After a task starts, it is not allowed to be preempted by other tasks
- (3) Each satellite ground station antenna can only establish a ground station to satellite link with one satellite at a time
- (4) Regardless of the data loss and link setup time during transmission
- (5) The satellite's power supply can meet the needs of satellite communications.

2.2. *Notations.* Sets and parameters are as follows:

- (i) $Task = \{task_1, task_2, task_3, \dots, task_n\}$: the set of data transmission tasks. For a task $task_i$, the following attributes are defined:
 - (a) et_i is the earliest available transmission time of $task_i$; it is determined by the end of data acquiring
 - (b) lt_i is the latest available transmission time of $task_i$; it is determined by the users who need data
 - (c) a_i is the amount of data; it depends on the process of data acquirement
 - (d) r_i is the transmission rate of $task_i$; it is determined by the satellite and the ground station to satellite link
 - (e) p_i is profit of $task_i$; it depends on the duration of the task, the importance, and the urgency.

The duration dur_i of task i can be determined by the amount of data and the transmission rate of the task i . The formula for duration is as shown in Equation (1):

$$dur_i = a_i/r_i. \quad (1)$$

- (ii) $TW = \{tw_1, tw_2, tw_3, \dots, tw_m\}$: the set of ground stations time windows. When a satellite passes over a ground station, it forms a time window. For a time window tw_j , the following attributes are defined:
 - (a) evt_j is the earliest visible time of time window j ; it is determined by the time when the satellite and a ground station can build a ground station to satellite link at the earliest
 - (b) lvt_j is the latest visible time of time window j ; it is determined by the time when the satellite and a ground station cannot continue to build a ground station to satellite link.

The variables are as follows:

- (i) $x_{i,j}$ is a decision variable. If time window j is chosen to transmit task i , then $x_{i,j}$ is 1. Otherwise, it is 0
- (ii) $s_{i,j}$ is the start time to transmit task i .

2.3. *Mathematical Model.* The planning and scheduling consist of two parts: to determine whether the task is executed in the task sequence and to select a suitable task execution time for the executed task. Such a task sequence has a profit value of its own. The main goal of the SDTS model is to determine a task sequence so that the profit value is as high as possible. The objective function of the model is as follows:

The objective function is

$$\max \sum_{i=1}^n \sum_{j=1}^m p_i x_{ij} \quad (2)$$

The constraints are

$$\sum_{i=1}^n \sum_{j=1}^m x_{ij} \leq 1, \quad (3)$$

$$et_i \leq s_i \leq lt_i, \quad (4)$$

$$s_i + dur_i \leq lt_i, \quad (5)$$

$$evt_j \leq s_i \leq lvt_j, \quad (6)$$

$$s_i + dur_i \leq s_{i+1}, i \in \{1, 2, 3, \dots, n-1\}, \quad (7)$$

$$x_{ij} \in \{0, 1\}. \quad (8)$$

Constraint (4) defines that the start time of the task cannot come earlier than the earliest available transmission time and cannot come later than the latest available transmission time. Constraint (5) defines that task i needs to be completed before the latest available transmission time. Constraint (6) defines the task needs to be executed within the visible time window of ground stations. Constraint (7) defines that task $i+1$ must start later than the end of task i . Constraint (8) defines the description of the range of values of the variables.

The above description of the model, assumptions, and constraints is the result of the simplification of the actual engineering problem, and the simplified SDTS problem still has a large search space. Designing an efficient solution algorithm is especially important to find a workable or satisfactory solution.

3. Heuristic Adaptive Large Neighborhood Search (H-ALNS) Algorithm

The adaptive large neighborhood search algorithm as a heuristic search algorithm uses the neighborhood destruction and reconstruction methods to continuously improve the search effect. As an improvement of the ALNS algorithm, the H-ALNS algorithm adds the content of heuristic task assignment method. After the new sequence structure is obtained, the execution time window and the position to execute each task are arranged by the heuristic method. The

adaptive mechanism is used to improve the probability of selecting a better performing operator. In the neighborhood destroy and repair method, H-ALNS designed a new method that is compatible with the SDTS problem. Below, the overall flow and some details of the H-ALNS will be introduced. First, the overall structure of the algorithm is described.

3.1. Algorithm Structure. The algorithm proposed includes generating the initial scheme, neighborhood search, neighborhood update, and adaptive mechanism. Similar to the ALNS, the algorithm includes two loop structures, and the local search method in the inner loop improves the neighborhood structure. The destroy method and the repair method are the main components of the local search. The destroy method destroys the existing neighborhood structure, causing the structure of the solution to be different from the initial solution. The repair method is used to readd the deleted partial solution to the set of solutions and generate a new neighborhood structure. The outer loop uses an adaptive mechanism to update the weight of the operator and determine the continued operation and termination of the algorithm. The flow chart of H-ALNS is shown in Figure 1.

As can be seen from Figure 1, the improvement of the ALNS algorithm mainly reflects in the addition of heuristic task assignment method, adaptive improvement of destroy method and repair method, and solution acceptance criterion in SDTS problem. The heuristic task assignment method is used to determine the time window and task execution time of each task after each large neighborhood search, which updates the order of task sequence. In the destroy method, this paper proposes random removal, time window ratio removal, duration removal, and remaining ground station resources removal. In the repair method, this paper proposes random insertion, earliest visible insertion, and latest end insertion. These methods of destroy and repair methods are closely related to the SDTS problem and can play an important role in improving the speed of optimization. After that, the pseudo code of H-ALNS, named Algorithm 1, is given as follows.

3.2. Heuristic Task Assignment. After getting a task sequence, a data transmission plan for this sequence needs to be determined. The heuristic task assignment method proposed in this paper is to arrange tasks. It determines the plan in a heuristic manner by the order of tasks obtained by H-ALNS. Task scheduling process consists of two parts: one is to select available time windows for a task to execute the task, and the other part is to select appropriate task execution time within the time window. Each task is arranged using two heuristic rules. In the process of selecting a time window for a task, the heuristic rule selects the time window in which the task can be completed earlier. In the process of selecting an execution time for the task, the position of time window in which the task can be executed is selected as the start time. Through heuristic task assignment, scheduled tasks will start and end as soon as possible at the earliest position of time window, leaving more available time window resources for subsequent tasks.

3.3. Generate the Initial Plan. The quality of the initial solution largely affects the subsequent optimization process. In the initial scenario, the set of solutions must be complete so that subsequent damage and repair can be performed. Therefore, we use the rules for selecting the earliest available location of the task to make a location selection for each data transmission task. There are two cases. The first case is when the task is inserted at the earliest selectable location, and there are no conflicts with other tasks. The task is directly inserted into the current location. In the second case, if the task is scheduled to collide with another existing task in the time window, the location where the current task can be inserted at the earliest available opportunity will be selected in all available time windows. The pseudo code generated by the initial scheme is shown in Algorithm 2.

3.4. Destroy Method. The destroy method refers to the destruction of the existing neighborhood structure, destroying the existing neighborhood structure and releasing the partial solution from the structure of the current solution. The released partial solution can reselect the position in the solution structure. The destroy method includes four kinds of destroy operators: random removal operator, time window ratio removal operator, duration removal operator, and ground station maximum resource residual removal operator.

3.4.1. Random Removal. The random method is to randomly select a task from a certain sequence and remove it from the original task sequence. This method is the simplest method to destroy the neighborhood structure, as shown in Figure 2. The arrows in the figure indicate randomly selected neighborhood destruction locations; in this example, randomly selected tasks 5 and 9 are deleted from the task sequence. The figure also shows the structure after the neighborhood change.

(1) Time Window Ratio Removal. The time window ratio method is determined according to the range of the available time windows within the duration window of the task, and the calculation method is shown in formula (9). For each task, according to the time window proportion ratio, the time window is selected to occupy a relatively large task for task culling.

$$\rho_i = \frac{a_i}{r_i(lt_i - et_i)}. \quad (9)$$

(2) Duration Removal. Duration refers to the duration of the task. It is determined by the amount of data in the task and the data transmission rate. When a task is rejected, tasks with a long duration are selected first. A task with a long duration will occupy more positions on the available time window than a short duration. When the long-time task is deleted, it will provide more available time windows for other tasks.

(3) Remaining Ground Station Resources Removal. The task sequence consumes different resources for satellite ground

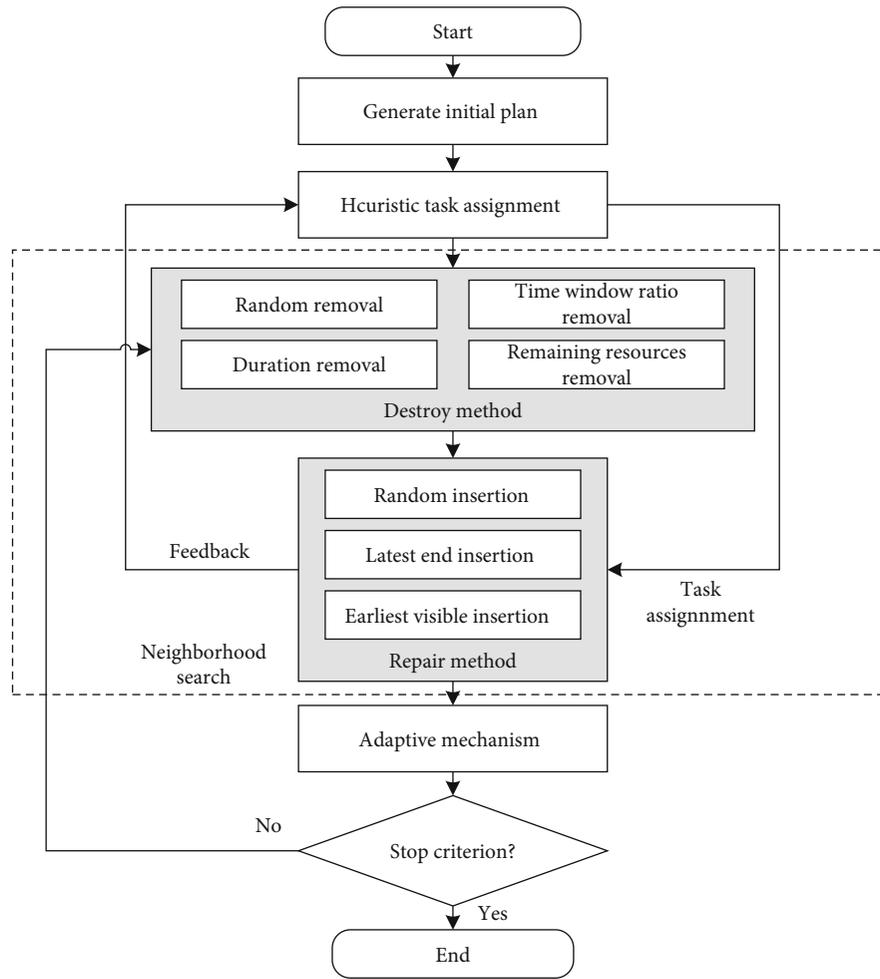


FIGURE 1: H-ALNS flow chart.

stations. Some ground station resources are used to a large extent, and some ground stations are assigned a small number of tasks. This result is practical. However, the level of usage imbalance is not ideal from the perspective of scheduling problems. Reasonable use of spare resources is something that needs to be considered during the scheduling process. According to the utilization of the ground station resources, the tasks in the ground station that select the remaining large resources are deleted from the task sequence.

3.5. Repair Method. The repair process is readded to the set of solutions by the appropriate location from the original solution set after the deletion method is executed, and the repair solution is obtained. Through the repair, a new neighborhood structure will be generated. The repair method defines three insertion operators: random insertion, the earliest visible insertion, and the latest end insertion operators. In this section, these three operators will be introduced individually.

3.5.1. Random Insertion. Select a position randomly in the destroyed solution set as the insertion position of the deleted task. The insertion of the random position can increase the diversity of solutions and will prevent the solution from falling into a local optimum. The process of random insertion is

shown in Figure 3. The deleted task 5 and task 9 are inserted between task 10 and task 2 by random selection, and the results after the repair are also shown in the lower part of the figure.

3.5.2. Earliest Visible Insertion. Select the earliest star visible position where the current task can start data transmission as the insertion position of the task. If the earliest position is not available, move the task backward to the position where the earliest task can be executed as the position where the task is inserted.

3.5.3. Latest End Insertion. Select the latest execution time allowed by the deleted task as the generation method for the new neighborhood structure. However, the insertion at the latest position also has the possibility of producing conflicts with existing tasks. If you encounter a conflict with a task, move the task forward to where insertion of the task is possible.

3.6. Adaptive Mechanism. The adaptive mechanism is an important part of the ALNS algorithm. Through self-adaptation, the algorithm can selectively bias the operator with good optimization effects, which makes the optimization effect

Input Plan to be planned $S_{initial}$, number of iterations P , number of stages Q , size of segment θ , operator weight ω_i , operator score σ_i , weight update parameter μ , acceptance parameter γ

Output Solution S

Generate initial plan S_0

Calculate profit of the plan

$S_{best} \leftarrow S_0$

For $i = 1$ to P

 Assign each task use heuristic task assignment method

 Select destroy operator, repair operator

$S'_i \leftarrow$ Destroy method(S_i)

$S_i \leftarrow$ Repair method(S'_i)

$Q = Q + 1$

If $S_i/S_{i-1} > 1$

$S_{best} \leftarrow S_i$ //Update current optimal solution

Else if $S_i/S_{i-1} < \theta$

$S_i \leftarrow S_{i-1}$ //Discard the current plan

End if

If $Q > \theta$

Update Operator weight ω_i

Update Operator score σ_i

Reset stages Q

End if

End for

Return $S \leftarrow S_{best}$

ALGORITHM 1: Heuristic adaptive large neighborhood search algorithm (H-ALNS)

Input Initial task sequence $S_{initial}$, available time window T

Output Planning S_0

For each task i in task sequence $S_{initial}$

 Select the earliest available location in T

If meet the constraints

 Insert task i at the current location

Else If available locations after the earliest location

 Select the top available location in T

 Insert the task i

Else go to the next task $\rightarrow i + 1$

End If

End For

Output planning S_0

ALGORITHM 2: Generate the initial plan

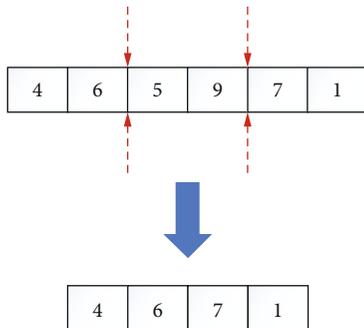


FIGURE 2: Random removal of two tasks.

more pronounced. First, parameters involved in the adaptive mechanism are needed to be defined, so that σ_i^d and σ_i^r are the score of the i th delete operator and the repair operator, respectively, and ω_i^d and ω_i^r are the weight of the i th delete operator and the repair operator, respectively. λ is the score added for each update process. The value is determined by the relationship between new solution and the current solution and the optimal solution. The score of the operator is updated after each deletion and repair process. The update principle is as follows:

- (i) When a new optimal solution is reached: $\sigma_i \leftarrow \sigma_i + \lambda_1$
- (ii) When the new solution is better than the current solution, but worse than the optimal solution: $\sigma_i \leftarrow \sigma_i + \lambda_2$
- (iii) When the new solution is worse than the current solution: $\sigma_i \leftarrow \sigma_i + \lambda_3$.

The θ iterations are set to be updated when number of stages Q reach the θ . And the number of stages will be reset. After the end of one stage, the weight of the operator is updated according to the score of the operator, and then the score of the operator is initialized. The weight update method is as follows:

$$\omega_i^d = (1 - \mu)\omega_i^d + \mu \frac{\sigma_i^d}{\sum_{i=1}^Q \sigma_i^d}, \quad (10)$$

$$\omega_i^r = (1 - \mu)\omega_i^r + \mu \frac{\sigma_i^r}{\sum_{i=1}^Q \sigma_i^r},$$

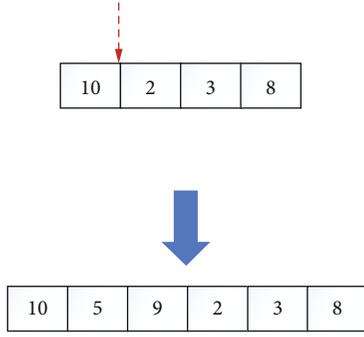


FIGURE 3: Random insertion of two tasks.

where $\mu \in [0, 1]$ is the weight update parameter, indicating the degree to which the operator is affected by the historical situation during the update process.

Using the roulette method to determine the method of selecting operators, the probability that each operator is selected is as follows:

$$\begin{aligned} p_i^d &= \frac{\omega_i^d}{\sum_{j=1}^Q \omega_j^d}, \\ p_i^r &= \frac{\omega_i^r}{\sum_{j=1}^Q \omega_j^r}. \end{aligned} \quad (11)$$

3.7. Solution Acceptance Criterion. Allow a solution that is worse than the current solution to be a new solution within an acceptable range by introducing an acceptance degree parameter γ . Compare the deleted and fixed solution with the previous round of solutions. If it is within the acceptable range, choose to accept the solution. If it exceeds the acceptable range, choose to abandon the solution.

3.8. Stop Criterion. The termination of the algorithm is based on the number of executions of the algorithm. When the number of executions of the algorithm reaches a preset maximum value, the algorithm terminates and outputs the result. When the size of the task sequence that needs to be optimized increases, the algorithm needs to execute more times to achieve the expected optimization effect. Therefore, the termination condition should be changed accordingly, which is compatible with the scale of the problem.

So far, we have described the overall process and the details of the algorithm in H-ALNS. Through repeated iterative optimization and adaptive adjustment of the operator, the quality of the solution is gradually improved, providing a practical application for short timeframes. This planning scheme can be used to effectively manage the operation of satellites and ground stations.

4. Computational Results

4.1. Experimental Scenario Design. The experiment we designed is consistent with actual satellite data transmission, and the capabilities of satellite and ground stations are similar to the real situation. The experimental scene was generated using the STK 9 tool. In terms of satellite missions, all

tasks are within a single task cycle. The number of experimental tasks includes different tasks ranging from small to very large. The number of tasks ranges from 50 to 1000. Among them, the simulation experiment of 50 to 300 tasks contains multiple scenes; the difference is that the time range allowed by the task is different. In other words, the distribution of tasks is different.

The verification of these tasks can fully reflect the performance of the scheduling algorithm in the presence of possibility. An example of five of these tasks is given in Table 1, and an example of five ground station time windows is given in Table 2.

The meaning of the satellite data transmission task and ground station time windows is the same as those indicated by the variables in Section 2.2.

4.2. Experimental Environment. The proposed algorithms are implemented by using MATLAB (2016b) on a laptop with a Core I5-3337U 1.8 GHz CPU, 4 GB memory, and the Windows 8.1 operating system.

4.3. Satellite Orbital Parameters. The experiments are based on several satellites in China. The location of a satellite in space is characterized by six orbital parameters: the length of the semimajor axis (LSA), eccentricity (E), inclination (I), argument of perigee (AP), right ascension of the ascending node (RAAN), and mean anomaly (MA). The initial orbital parameters for the satellite are presented in Table 3.

4.4. Comparison Algorithm. We have selected five contrast algorithms to verify the effect of our proposed H-ALNS algorithm under the same simulation conditions. These algorithms are always used in solving the SDTS problem. The five algorithms selected include a metaheuristic algorithm and four heuristic algorithms: **Algorithm 1**: standard genetic algorithm (GA); **Algorithm 2**: higher task profit first (HPF); **Algorithm 3**: earlier allowed tasks first priority (EATF); **Algorithm 4**: higher task unit time yield first (HUPF); and **Algorithm 5**: shorter task duration first (SDF). Due to the large task size of the SDTS problem, it is difficult to obtain a satisfactory solution using the exact solution algorithm in a limited time.

4.5. Evaluation Index. The profit value of the task sequence and the completion rate of the task are used as criteria for evaluating the performance of an algorithm. The profit value provides a good evaluation in the situation when the task sequence can achieve the planning goal. The task completion rate is considered using the actual application process. Additionally, sufficiently meeting the user's demand is a matter of concern.

Large neighborhood search algorithm parameters are as follows:

- (i) The score increment: $\lambda_1 = 30, \lambda_2 = 20, \lambda_3 = 10$
- (ii) The score of destroy method: $\sigma_1^d = \sigma_2^d = \sigma_3^d = \sigma_4^d = 100$
- (iii) The score of repair method: $\sigma_1^r = \sigma_2^r = \sigma_3^r = 100$
- (iv) Size of segment: $\theta = 50$ iterations

TABLE 1: Example of satellite data transmission task.

ID	et	lt	a	r	P
1	19	10818	14	1	78
2	523	8527	7	1	132
3	781	7361	26	2	147
4	1018	11658	12	1	95
5	1726	20164	18	2	125

TABLE 2: Example of satellite ground station time window.

Ground station name	ID	evt	lvt	tw
Ground station 1	1	4895	5331	436
	2	0	254	254
Ground station 2	3	0	421	421
	4	10746	11322	576
	5	32759	32967	208

(v) Weight update parameter: $\mu = 0.5$

(vi) Maximum number of iterations: 5000

(vii) Acceptance parameter: $\gamma = 0.8$.

First, we compare the effectiveness of the task planning profit of different algorithms in different task scenarios. The results are shown in Table 4, where each row represents a different algorithm planning result in the same task scenario. The task size can be divided into three types: small-scale, medium-scale, and large-scale task scenarios. Under each task scale, different algorithms have their own approximate optimization scope, which reflects the planning gap of the algorithms in the same task scenario.

In the scenario, the proposed H-ALNS and GA, as meta-heuristic algorithms, perform significantly better than the four heuristics. This is because the metaheuristic algorithm is designed with a more complex sequence adjustment mechanism, which prevents the algorithm from falling into a local optimal situation in a random way. In the scenario of 50-2, H-ALNS only got a profit of 2654.7, which is much lower than profits of the other two scenarios. This is because the profit of some tasks is low. It can be seen that from the number of tasks completed in this scenario, 49 tasks are completed. The method of jumping out of the local optimum produces an advantage in large-scale scenarios, such as the 250 and 300 tasks. Among the four heuristic algorithms, ETF and HUPF can get higher profits. The idea of ETF is similar to the idea of heuristic task arrangement, which also proves to some extent that heuristic task arrangement can choose the appropriate execution position for each task. Compared with small-scale task scenarios, the algorithms in large-scale scenarios have more local optimality. The random neighborhood destruction and repair methods in H-ALNS possibly create the seemingly stable optimization and break. Although the solution obtained temporarily has no current solution, it can achieve better performance through repeated optimization. In scenarios of 250 and 300 task, H-ALNS optimization effect is better than the other five comparison

algorithms. Among these four heuristic algorithms, SDF algorithm can get the highest profit. In different scenarios of each task size, the algorithms' effects are tested according to the different distribution of tasks. It can be seen from the results that H-ALNS has a good optimization effect under different task scales and different distribution of tasks.

In many cases for practical engineering applications, larger-scale scenarios are encountered. The larger number of tasks is accompanied by an increase in conflicts between tasks and a lack of resources available for the tasks. We verified satellite data transmission planning problems in large-scale scenarios. The results are shown in Table 5. As seen from Table 5, for the range of 400 to 1000 tasks, the profit increase when compared to the four heuristic algorithms is not as distinct as it was with 300 tasks. This is because the initial task sequence is often given. For conflicts between tasks that are difficult to reduce, the H-ALNS and GA algorithms can eliminate many conflicts by constantly updating the structure of the solution. However, the update of the operator weight is lacking in the GA algorithm, and it is difficult to drive the result toward the direction of faster optimization. Therefore, the profit gap between the two algorithms increases further with the increase in the number of tasks. As the number of tasks increased, the overall performance of profits showed an upward trend, with a slight decline under 500, 600, and 1000 tasks. The decline in 500 and 600 task scenarios is related to the distribution of tasks. Many tasks require completion within a time range, which leads to the conflict between tasks. Some of conflicts cannot be reduced, and some tasks have to be abandoned. When the number of tasks increases from 800 to 1000, the profit increases first and then decreases. This is because the available time window resources of ground stations are limited, and the total amount of tasks that can be completed is also limited.

In order to verify the optimization effect of H-ALNS, evaluation is performed using $Dev.$, proposed by Pezzella [17]. $Dev.$ is an indicator for calculating the difference between the two algorithms. The calculation formula is as shown in Equation (12):

$$Dev. = \left(\bar{P}_{H-ALNS} - \bar{P}_{comparison\ algorithm} \right) / \bar{P}_{comparison\ algorithm} \times 100\% \quad (12)$$

The calculation results of $Dev.$ is shown in Figure 4. In Figure 4, the numerical number represents the position of the algorithm, and the order is the same as the order in the result table above, such as no. 1 for H-ALNS and no. 2 for GA. It can be seen from the results that the gap between GA and H-ALNS is within 50%, and within the range of 400 to 600 tasks, the gap between the two is small. In the process of increasing the scale of tasks from 700 to 1000, the gap between different algorithms has a certain degree of fluctuation, but the overall gap is larger than the difference between 400 and 600 tasks. This reflects the better optimization performance of H-ALNS in large-scale problem optimization.

TABLE 3: Satellite orbital parameters.

Parameter	LSA	E	I	AP	RAAN	MA
Value	7141701.7	0.000627	98.5964	95.5069	342.307	125.2658

TABLE 4: Results of profit under small-scale examples.

Instance	H-ALNS	GA	HPF	EATF	HUPF	SDF
50-1	4169.9	2417.6	2013.2	1801.0	2287.8	2210.4
50-2	2654.7	2064.7	972.5	1203.2	1142.8	1305.5
50-3	3173.2	2454.5	1354.7	1407.1	1493.9	1630.6
100-1	6157.1	5452.2	3281.6	3395.1	2630.4	4345.1
100-2	5593.4	4402.2	2749.0	2243.5	2561.1	4213.8
100-3	4974.4	3989.2	2707.2	3060.6	2707.0	4691.4
150-1	6335.3	5177.3	3603.4	2388.6	3261.6	5142.3
150-2	8705.7	5776.6	3676.4	3218.8	3890.5	5744.1
150-3	6314.8	4789.4	2940.8	3103.5	2996.2	3764.4
200-1	10116.5	6389.4	4416.4	3550.9	3731.2	6331.2
200-2	8519.3	6509.3	4766.0	5177.2	4636.2	3951.8
200-3	9382.7	7552.7	3318.0	4246.4	3804.8	4228.2
250-1	9093.4	7209.9	4082.1	3039.2	5449.3	5913.4
250-2	11244.3	8958.2	4414.0	4304.4	4310.4	5505.8
250-3	10082.0	8990.3	5268.2	4344.7	4680.1	5135.9
300-1	13476.9	8783.5	4501.4	5966.2	4374.7	4844.3
300-2	11377.7	9231.0	4960.3	3828.1	5692.4	6210.4
300-3	10878.4	7758.1	4960.3	3828.1	5692.4	6831.4

TABLE 5: Results of profit under large-scale examples.

Instance	H-ALNS	GA	HPF	EATF	HUPF	SDF
400	11438.3	9729.7	5651.3	3390.5	4938.7	7051.8
500	10284.4	8454.1	4917.4	3590.1	6172.3	7247.4
600	10728.6	9681.9	5620.3	4548.3	5343.7	8124.1
700	13965.5	9153.1	6371.2	3990.6	6098.1	7045.7
800	13552.4	10458.6	6012.3	5028.7	6892.6	7931.6
900	15075.1	9860.2	6280.2	4390.4	7119.2	8892.2
1000	14588.9	11228.9	7091.7	5508.2	6758.1	9022.9

Among these four heuristic algorithms, the difference between SDF and H-ALNS is the smallest, and volatility in different scenarios is small. It can also be seen from the results that ETF algorithm is a heuristic algorithm with the largest gap, which indicates that the relation between tasks' start time and the optimization benefit is small. The most unstable of four heuristic algorithms is the ETF, and it is also the worst-optimized algorithm among these six algorithms.

After using profit, the number of task completions is analyzed as an evaluation, and the results are shown in Table 6. When there are only 50 tasks in the planning scenario, the difference in the number of tasks completed by different optimization algorithms is small, and even the heuristic algorithms can complete the same number of tasks in the scenario as metaheuristic algorithms. In the scenario of

50-1, H-ALNS, GA, and HUPF can get the same number of 47 tasks that can be completed, and the task completion rate reaches 94%. When the task size is increased from 50 to 100, the gap between the algorithms begins to appear. The most obvious gap is between the H-ALNS algorithm and the EATF algorithm. With a further increase in the task scale, the gap between the algorithm planning results is more obvious, and the resource limitations add additional requirements for the algorithm. When the scale of tasks is 200, in the scenarios of 200-2 and 200-3, HUPF has a similar task completion rate that is very close to H-ALNS and GA algorithms. Although the number of executable tasks after algorithm optimization increases with the increase of the task size, the task completion rate shows a downward trend due to the limitation in capabilities. In all experimental scenarios, the number of tasks completed by the GA algorithm in three scenarios is the same as H-ALNS. In other 15 scenarios, H-ALNS has the largest number of tasks completed.

Subsequently, we analyze the comprehensive performance of different algorithms in different task scale scenarios, while still comparing the task planning profit and the number of task completions. The results are shown in Figures 5 and 6. The plan profit and task completion numbers are the average performance of the algorithms in each scenario and at a specific task scale, obtained using an average calculation method.

From the perspective of the task profit for different task scales, H-ALNS performs best, followed by GA. The superiority of these two algorithms is more obvious than the differences in the others. Among the other four algorithms, the performance of the SDF algorithm is the best, and it has certain profit advantages when compared with the other three heuristic algorithms. The performance of the H-ALNS algorithm and the performance of the other algorithms are accompanied by an increase in the number of tasks, and the profit gap is more obvious. When the scale of tasks is 300, H-ALNS has an increase of 19.78% in the task sequence compared to GA. At this time, H-ALNS can accomplish 1.46% higher compared to GA in terms of the number of tasks completed. From the point of view of the number of tasks completed, the number of tasks that can be successfully executed after an increase in the task size increases at the same time. However, the increase rate shows a downward trend between algorithms, and accordingly, the completion rate of the task can be increased in the number of accompanying tasks. The total process presents a downward trend.

The above analysis shows that the proposed H-ALNS algorithm offers better performance than the other metaheuristic algorithm, GA, and the four heuristic algorithms, both in terms of task sequence profit and task completion. Among these four heuristic algorithms, SDF has the best performance in terms of task profit. In terms of tasks completed, the number of data transmission tasks that can be completed

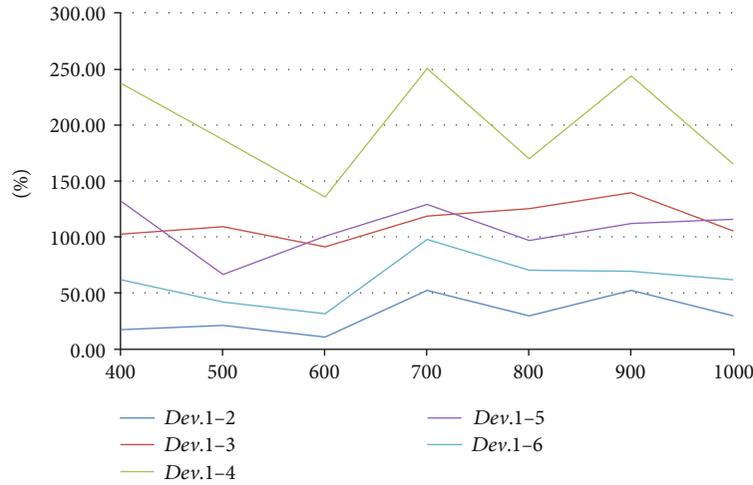


FIGURE 4: Results of Dev.

TABLE 6: Task completion number.

Instance	H-ALNS	GA	HPF	EATF	HUPF	SDF
50-1	47	47	45	45	47	46
50-2	49	46	48	45	48	47
50-3	48	48	47	46	47	46
100-1	96	93	92	90	94	90
100-2	97	94	92	91	95	93
100-3	94	90	88	89	91	90
150-1	143	139	135	137	138	135
150-2	138	137	134	132	135	133
150-3	141	138	136	133	137	135
200-1	186	179	178	175	180	178
200-2	185	182	182	177	184	182
200-3	184	184	183	180	183	181
250-1	237	234	230	230	233	232
250-2	235	233	231	228	232	231
250-3	233	229	226	225	227	225
300-1	279	278	268	272	275	270
300-2	277	274	265	270	272	268
300-3	279	271	271	269	270	269

using HUPF algorithm is the most. Meanwhile, the H-ALNS algorithm can obtain better planning results in medium-scale and large-scale task scenarios, which also reflects that the design of the destruction method and the repair method is more suitable for scenarios with larger task sizes. In addition, we also verified the effect of the size of the weight update parameter in the algorithm on the result. We take the scenario of 400 to 1000 tasks as an example. The results are shown in Table 7.

The results are shown in Table 7. The results show that the optimization effect of H-ALNS is most ideal when the weight update parameter is set to 0.5, in addition to the

income level that can be obtained when the weight is increased or decreased. When the scale of tasks is 800 and 1000, and the operator weight is 0.9, it can obtain the second-best optimization effect. This is related to conflicts of tasks. It needs to be affected by historical information as little as possible, and the weight of the destroy method and repair method needs to be adjusted according to the process of the neighborhood search. It is worth mentioning that we designed the case where the parameter is 0, that is, the weight is not updated. The result of the planning is the worst of all of the weight update parameters, thus proving that a nonzero design update parameter is necessary.

4.6. Through Experimental Analysis, the following Conclusions Can Be Made. H-ALNS and GA can achieve better profit results and task execution results than the four heuristic algorithms. Among all of the algorithms investigated, the algorithm of H-ALNS is the best. This can be seen from both the task completion including profit and number of tasks completed, and the Dev. indicators. The experiments we designed cover scenarios of different task sizes. From the results, H-ALNS is the most prominent when looking at large-scale scenarios

Among the four heuristic algorithms, SDF performs best in terms of task profit when compared to the other three algorithms. Short task durations provide opportunities for more tasks to be accommodated in the sequence. In terms of task completion, HUPF is the best of the heuristic algorithms

It is reasonable to set the update weight parameter to 0.5 for H-ALNS. In the experimental verification, the profit of the planning schemes generated by other parameter settings was not as high as when the parameter is set to 0.5. In addition, considering the special case where the parameter is set to 0, that is, the weight value is not changed during the optimization process, the profit result is the lowest. A reasonable setting of the weight update parameters greatly improves the effect of algorithm optimization.

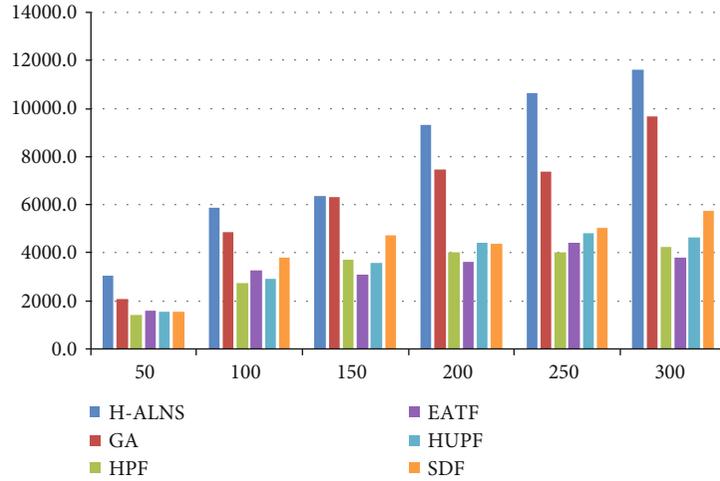


FIGURE 5: Average profit performance.

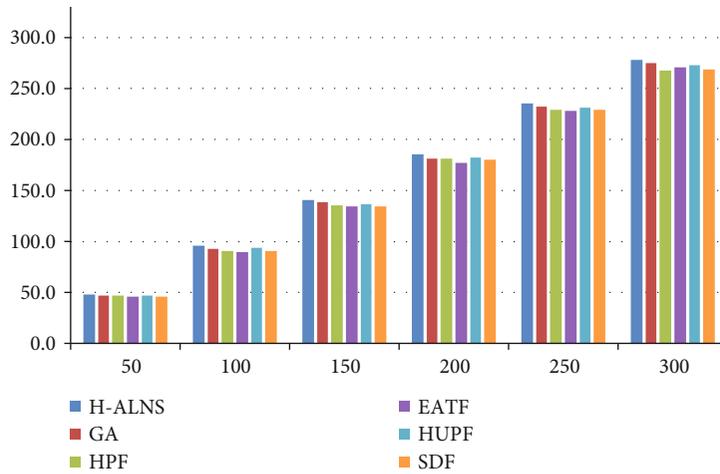


FIGURE 6: Average task competition performance.

TABLE 7: Profit under different weight update parameters.

Instance	$\mu = 0.1$	$\mu = 0.3$	$\mu = 0.5$	$\mu = 0.7$	$\mu = 0.9$	$\mu = 0$
400	9389.6	9827.4	11438.3	10274.5	9568.3	8753.4
500	9064.7	9421.8	10284.4	9621.9	9378.5	7285.1
600	9073.8	9601.1	10728.6	9507.2	9381.4	8219.5
700	11527.3	12759.2	13965.5	13684.3	12985.6	9483.2
800	11962.5	12894.1	13552.4	12763.8	13074.7	10184.6
900	13752.6	14864.2	15075.1	14719.4	13966.3	11022.7
1000	12836.7	12905.3	14588.9	13174.3	13286.5	11795.3

5. Conclusion

In this paper, the satellite data transmission problem of multiple satellites and ground stations is studied. Based on the analysis of the difficulties and characteristics of the problem, the mixed integer model of SDTS is constructed, and the assumptions and constraints are proposed. After that, we designed an adaptive large neighborhood search algorithm to solve the

SDTS problem. The algorithm designs a heuristic rule to generate the initial plan. At the same time, the algorithm includes four destroy methods and three repair methods. The adaptive mechanism is introduced into the H-ALNS to improve the optimization effect and jump out of the local optimum. The main idea of the algorithm is to continuously update the neighborhood structure through destruction and repair. The operation operator updates the operator weight according to the optimization effect in an iterative optimization process to ensure an efficient search and improve the planning effect.

In the experimental results, small-scale and large-scale examples demonstrate that the heuristic algorithms perform well when considering both the profit and the number of task completions. However, when compared to the heuristic algorithms, H-ALNS has an even more desirable performance. Through experimental verification, we can prove that heuristic algorithms, the destruction method, the repair method, and the adaptive mechanism designed in the algorithm can solve the SDTS problem. Especially in the case of large-scale tasks, the performance of our H-ALNS algorithm is superior.

In future research, the algorithm operator design, adaptive mechanism design, and improvement strategy after each repair will be further studied. At the same time, using parallel computing as an efficient calculation method will also be considered in the design process of the algorithm. As our next step, we will design an application system to put our proposed algorithm into actual use.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare no conflict of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant 71501179 and 71501180. Thanks to the reviewers for reading my manuscript and for their valuable comments on the article. I also thank my classmates and teachers for their help with the research work.

References

- [1] K. J. Zhu, J. F. Li, and H. X. Baoyin, "Satellite scheduling considering maximum observation coverage time and minimum orbital transfer fuel cost," *Acta Astronautica*, vol. 66, no. 1-2, pp. 220-229, 2010.
- [2] X. Fatos, S. Junzi, B. Admir, B. Alexander, and B. Leonard, "Genetic algorithms for satellite scheduling problems," *Mobile Information Systems*, vol. 8, no. 4, pp. 351-377, 2012.
- [3] V. Kolicic, X. Herrero, F. Xhafa, and L. Barolli, "Local Search and Genetic Algorithms for Satellite Scheduling Problems," in *2013 Eighth International Conference on Broadband and Wireless Computing, Communication and Applications*, Compiegne, France, 2013.
- [4] X. Chu, Y. Chen, and L. Xing, "A branch and bound algorithm for agile earth observation satellite scheduling," *Discrete Dynamics in Nature and Society*, vol. 2017, Article ID 7345941, 15 pages, 2017.
- [5] Y. Song, B. Song, Z. Zhang, and Y. Chen, "The satellite downlink replanning problem: a BP neural network and hybrid algorithm approach for IoT internet connection," *IEEE Access*, vol. 6, pp. 39797-39806, 2018.
- [6] Y. Song, X. Ma, Z. Zhang, L. Xing, and Y. Chen, "A hybrid dynamic population genetic algorithm for multi-satellite and multi-station mission planning system. Bio-inspired Computing: Theories and Applications," in *BIC-TA 2018*, Springer, 2018.
- [7] F. Xhafa, J. Sun, A. Barolli, M. Takizawa, and K. Uchida, "Evaluation of Genetic Algorithms for Single Ground Station Scheduling Problem," in *2012 IEEE 26th International Conference on Advanced Information Networking and Applications*, Fukuoka, Japan, 2012.
- [8] H. Renjie, B. Baocun, C. Yingwu, and T. Yuejin, "Multi-satellite mission planning for environmental and disaster monitoring satellite system," in *SpaceOps 2008 Conference*, Heidelberg, Germany, 2008.
- [9] Y. He, L. Xing, and Y. Chen, "Software design of autonomous mission planning for new imaging satellite," in *SpaceOps 2016 Conference*, Daejeon, Korea, 2016.
- [10] D. Karapetyan, S. Mitrovic Minic, K. T. Malladi, and A. P. Punnen, "Satellite downlink scheduling problem: a case study," *Omega*, vol. 53, pp. 115-123, 2015.
- [11] L. He, X. Liu, G. Laporte, Y. Chen, and Y. Chen, "An improved adaptive large neighborhood search algorithm for multiple agile satellites scheduling," *Computers & Operations Research*, vol. 100, pp. 12-25, 2018.
- [12] X. Liu, G. Laporte, Y. Chen, and R. He, "An adaptive large neighborhood search metaheuristic for agile satellite scheduling with time-dependent transition time," *Computers & Operations Research*, vol. 86, pp. 41-53, 2017.
- [13] S. Ropke and D. Pisinger, "An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows," *Transportation Science*, vol. 40, no. 4, pp. 455-472, 2006.
- [14] E. Demir, T. Bektas, and G. Laporte, "An adaptive large neighborhood search heuristic for the pollution-routing problem," *European Journal of Operational Research*, vol. 223, no. 2, pp. 346-359, 2012.
- [15] V. C. Hemmelmayr, J. Cordeau, and T. Crainic, "An adaptive large neighborhood search heuristic for two-echelon vehicle routing problems arising in city logistics," *Computers & Operations Research*, vol. 39, no. 12, pp. 3215-3228, 2012.
- [16] B. Li, D. Krushinsky, T. Van Woensel, and H. A. Reijers, "An adaptive large neighborhood search heuristic for the share-a-ride problem," *Computers & Operations Research*, vol. 66, pp. 170-180, 2016.
- [17] F. Pezzella, G. Morganti, and G. Ciaschetti, "A genetic algorithm for the flexible job-shop scheduling problem," *Computers and Operations Research*, vol. 35, no. 10, pp. 3202-3212, 2008.