

Research Article

Robust Satellite Scheduling Approach for Dynamic Emergency Tasks

Xuejun Zhai,^{1,2} Xiaonan Niu,^{1,2} Hong Tang,^{1,2} Lixin Wu,³ and Yonglin Shen⁴

¹State Key Laboratory of Earth Surface Processes and Resource Ecology, Beijing Normal University, Beijing 100875, China

²Key Laboratory of Environment Change and Natural Disaster, Ministry of Education, Beijing Normal University, Beijing 100875, China

³School of Environment Science and Spatial Informatics, China University of Mining and Technology, Xuzhou 221116, China

⁴Faculty of Information Engineering, China University of Geosciences, Wuhan 430074, China

Correspondence should be addressed to Hong Tang; hongtang@bnu.edu.cn

Received 30 June 2015; Accepted 15 October 2015

Academic Editor: Xiaobo Qu

Copyright © 2015 Xuejun Zhai 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.

Earth observation satellites play a significant role in rapid responses to emergent events on the Earth's surface, for example, earthquakes. In this paper, we propose a robust satellite scheduling model to address a sequence of emergency tasks, in which both the profit and robustness of the schedule are simultaneously maximized in each stage. Both the multiobjective genetic algorithm NSGA2 and rule-based heuristic algorithm are employed to obtain solutions of the model. NSGA2 is used to obtain a flexible and highly robust initial schedule. When every set of emergency tasks arrives, a combined algorithm called HA-NSGA2 is used to adjust the initial schedule. The heuristic algorithm (HA) is designed to insert these tasks dynamically to the waiting queue of the initial schedule. Then the multiobjective genetic algorithm NSGA2 is employed to find the optimal solution that has maximum revenue and robustness. Meanwhile, to improve the revenue and resource utilization, we adopt a compact task merging strategy considering the duration of task execution in the heuristic algorithm. Several experiments are used to evaluate the performance of HA-NSGA2. All simulation experiments show that the performance of HA-NSGA2 is significantly improved.

1. Introduction

Earth observing satellites (EOSs), orbiting the Earth, are able to collect images of specified areas of the Earth's surface at the request of customers by using observation sensors [1]. EOSs not only play key roles in application for environment surveillance, reconnaissance, resource investigation, and other fields, but also support many important services, such as remote sensing, navigation, geodesy, and monitoring [2]. As a result, many countries, especially China, tend to increase their investments to develop associated techniques and EOSs. For example, China plans to launch four small optical satellites and four small SAR satellites to form a natural disaster-monitoring constellation [3]. Although the number of EOSs is continuously increasing, they are still unable to satisfy the requirements of various customers [4]. Therefore, it is very important to develop effective methods of satellites observation scheduling to make full use of scarce

satellite resources to better satisfy the demands of customers. The satellite observation scheduling problem (SOSP) is to reasonably assign satellite resources and time windows to Earth observation tasks on the precondition of satisfying complex constraints [5]. Additionally, SOSP can be seen as a kind of multidimensional knapsack problem that is NP-hard [6].

From the previous survey, the satellite observation tasks scheduling problem can be divided into two classes, that is, static tasks scheduling and dynamic tasks scheduling. Over the past decades, the satellite scheduling imaging problem has been intensively investigated [7–10]. Regarding static single satellite scheduling, Lin et al. employed mathematical programming methods such as Tabu search, Lagrange relaxation, and liner search to solve the scheduling problem and acquired a near-optimal schedule [11–13]. Wolfe and Sorensen proposed three corresponding algorithms including a dispatch algorithm, a look-ahead algorithm, and a genetic algorithm to

model the imaging satellite scheduling problem [14]. Gabrel and Murat presented two upper bound procedures based on the graph theory and a column generation technique [15]. Vasquez and Hao formulated the satellite observation scheduling problem into a generalized version of the well-known knapsack model and used the Tabu search method to solve the problem [16]. Furthermore, they proposed a partition-based approach UPPB to obtain tight upper bounds to evaluate the TS search algorithm [17]. Several exact methods, such as depth first branch and bound or Russian dolls search and some approximate methods, such as greedy search and Tabu search, were used by Bensana et al. to solve the scheduling problem of the SPOT5 satellite. They viewed the scheduling problem as a value constraint satisfaction problem or an integer linear programming problem [6].

Although the single satellite scheduling problem can be solved perfectly by the above methods, with the development of the EOS system, these methods are unable to satisfy the needs of multisatellite system scheduling. In later studies, many researchers adopted a heuristic algorithm to solve the multisatellite scheduling problem such as Zweben et al. [18], Frank et al. [19], Bianchessi and Righini [20], Wang and Reinelt [21], and Marinelli et al. [22]. Bianchessi and Righini proposed a FIFO heuristic and a look-ahead insert heuristic algorithm to solve the scheduling problem [20]. Frank et al. used a constraint based interval planning framework to model the problem and proposed a heuristic for guiding this search procedure. Wang and Reinelt considered image downloads and proposed a priority-based conflict-avoided heuristic algorithm [21]. With the development of the intelligent optimization algorithm, many researchers proposed using intelligent algorithms to solve the complex scheduling problem. The intelligent algorithms mainly include genetic algorithm and ant colony optimization algorithm. Barbulescu et al. [23], Wang et al. [24], Baek et al. [25], Chen et al. [26], and Mansour and Dessouky [7] used the genetic algorithm (GA) to address the satellite scheduling problem. Barbulescu et al. proposed Genitor, a genetic algorithm, and proved that it performs well for a broad range of problem instances [23]. A multiobjective earth observation satellite scheduling method called SPEA2, based on the strength of the Pareto evolutionary algorithm, was proposed by Wang et al. [24]. Additionally, Mansour and Dessouky solved the satellite scheduling problem by developing a genetic algorithm considering two objectives, that is, the profits and the number of acquired photographs [7]. Ant colony optimization (ACO) was also used to address this problem. Based on ant colony optimization, Zhang et al. [27] and Liu et al. [28] considered multisatellite resources and task merging, respectively. Wu et al. presented a novel two-phase based scheduling method with the consideration of task clustering [29]. Moreover, Qiu et al. proposed the first finish first schedule with discard task moving back (FFFS-DTMB) and accommodate discard task predicting coexistence with task moving back (ADTPC-DTMB) algorithms to solve the problem [30]. Globus et al. developed an evolutionary algorithm to solve this problem and compared it with existing algorithms [31, 32]. From the survey of static tasks scheduling, we find that the heuristic algorithm is shown to perform well for simple problems and

the genetic algorithm is proven to be suitable for large scale, more complex problems [23].

Static scheduling assumes that all imaging tasks have been submitted before scheduling and once the scheduling scheme is produced, it is immutable until all tasks have been finished. In practice, because emergent events usually occur unexpectedly, with uncertainties of occurrence time and number, traditional static scheduling cannot attend to these emergent tasks in time. It is suggested that dynamic scheduling copes with these unexpected factors.

Until now, there are a few research efforts towards the dynamic scheduling on Earth observation satellites. Pemberton and Greenwald described the problem of dynamic scheduling and discussed contingency conditions [33]. Kramer and Smith proposed a repair-based search method for the oversubscribed scheduling problem [34]. Verfaillie and Schiex developed an approach employing local adjustment to solve the dynamic constraint satisfaction problem (DCSP) [35]. Considering the tradeoff between the performance and degree of adjustment, Wang et al. proposed a rule-based heuristic algorithm to solve the dynamic scheduling problem [36]. Also, Wang et al. established a multiobjective mathematic programming model for the dynamic real-time scheduling of EOSs and proposed a dynamic real-time scheduling algorithm called DMTRH to solve the problem [37]. Through analyzing the main constraints, Dishan et al. constructed an integer programming model and proposed a Rolling Horizon (RH) strategy with a mixed triggering mode to schedule the dynamic tasks and common tasks together [2].

To the best of our knowledge, EOSs play an important role in the process of disaster relief. For example, when an earthquake occurs, new emergency tasks will be submitted to the satellite observation system to acquire images that contain disaster information. According to the needs of disaster relief, more and more different emergency tasks will arrive. So far, just a few dynamic scheduling methods consider dealing with the emergency tasks. After analyzing the dynamic properties of satellite scheduling, Wang et al. proposed an optimization model and two heuristic algorithms to solve the problem [38]. However, they did not consider the task merging method in the heuristic algorithms. Wu et al. adopted the ant colony optimization plus iteration local search (ACO-ILS) approach to resolve the multisatellite emergency tasks and common tasks scheduling problem [4]. However, they did not account for the uncertainty of emergency tasks. Moreover, Wang et al. presented a dynamic scheduling algorithm called TMBSR-DES, which comprehensively considers task merging, backward shift, and rehabilitation [39]. The task merging strategy in TMBSR-DES only combines new tasks. The robustness of schedule has not been considered in all of the above algorithms.

To sum up, the studies on dynamic emergency scheduling mentioned above have the following shortages.

- (1) Some dynamic scheduling methods ignored the timeliness of emergency tasks.
- (2) The real-time scheduling method [2, 37, 38] took into account the timeliness of dynamic emergency tasks,

but it did not consider the influence of the common tasks on satellites.

- (3) Most of studies overlooked the robustness of dynamic satellite scheduling.

In this paper, we propose a robust scheduling approach oriented to both dynamic tasks and common tasks. We establish a robust satellite scheduling model to maximize the profit and robustness of the schedule. To solve the model, we design a combined algorithm HA-NSGA2 that is made up of multiobjective genetic algorithm NSGA2 and rule-based heuristic algorithm. NSGA2 is to acquire a flexible and highly robustness initial schedule in the scheduling of common tasks. Then a heuristic algorithm is designed to insert more emergency tasks dynamically. Meanwhile, to improve the revenue and resource utilization, we adopt a compact task merging strategy considering the duration of task execution in the heuristic algorithm. Based on the heuristic algorithm, NSGA2 is employed to optimize the solution so as to keep high continuous robustness for new emergency tasks.

The major contributions of this paper are summarized as follows. (1) The robustness is considered in the whole process of common tasks and dynamic emergency tasks scheduling to improve the scheduling ability of resisting disturbances. (2) To leave more opportunities for emergency tasks and improve the revenue and resource utilization, a compact task merging strategy is used. (3) Through considering the advantages of the heuristic algorithm and NSGA2 comprehensively, we propose an efficient solution to deal with the emergency tasks dynamically.

The remainder of this paper is organized as follows. The characteristics of emergency tasks and the solution framework are described in Section 2. In Section 3, we present the robust scheduling model. We introduce the corresponding solution to the model including the multiobjective genetic algorithm, rule-based heuristic algorithm, and neighborhood operator in detail, in Section 4. The simulation results and performance analysis, through comparison with other methods, are given in Section 5. Section 6 concludes the paper with some future research directions.

2. Problem Description

Earth observing satellites (EOSs), orbiting the Earth, collect images of the Earth's surface. As shown in Figure 1(a), a strip of EOSs can be formed on the ground by the subsatellite point of satellite and the field of view of the sensor, slewing angle of the sensor, and observation duration. Satellite mission scheduling allocates the limited satellite resources to observing tasks efficiently and reasonably. Figure 1(b) shows that each schedule is a sequence of tasks ordered in time for an EOS.

The EOSs work in a complex environment in which there exist many dynamic factors, as follows:

- (i) The change in task properties: the change in user requirements or the reasonable task attributes will lead to a change in task priority. When the priority of the task changes, we cancel the task before it is

performed. Then the task, which is viewed as an emergency task, is added to the emergency task set.

- (ii) The change in the state of satellites: sometimes, certain satellites may be out of use because of malfunction, memory, or energy shortage, and so on. As a result, the initial schedule cannot be executed continuously. These tasks which are affected by the satellites can be seen as emergency tasks to be inserted.
- (iii) The arrival of emergency tasks: according to the actual requirement, users may insert some new tasks when a scheduling scheme is executed. Specifically, there are some new incoming emergency tasks caused by emergent events, for example, earthquakes, fire, and landslides. Take an earthquake as an example. There is an urgent need to reasonably utilize the existing satellites to rapidly image the affected area during a short time period. Once the emergency tasks are submitted and scheduled, EOSs can acquire remote sensing images of disaster area quickly, which can provide rapid and effective information for quick investigation and assessment of earthquake damage.
- (iv) Uncertainty of weather conditions: some imaging tasks may not be completed because of a change in weather conditions. To satisfy user requirements, these tasks must be rearranged.

The four cases mentioned above can be seen as emergency tasks to be inserted into a scheduling scheme. Thus, the dynamic scheduling problem can be described as a unified form of inserting new tasks. In this paper, we research dynamic scheduling focus on new incoming emergency tasks. The general method is to produce a temporary schedule and then to adjust the schedule as quickly as possible. When dynamically adjusting the initial schedule, the solution stability is an important problem. In fact, the satellite application is a complex process and the orders can only be uploaded with the special equipment within limited visible time windows, so excessive changes may cause great operational trouble [36]. Hence, we should not only consider arranging more emergency tasks dynamically but also decrease the disturbances to the initial schedule as much as possible. Moreover, the uncertainty of arrival time is an important property of new tasks. Thus, we should consider the timeliness of emergency tasks in dynamic scheduling. Therefore, the goal of our research is to arrange emergency tasks as more as possible in smaller disturbances, considering the timeliness of the emergency tasks. As a result, the optimization objectives in this paper are revenue and robustness.

In this paper, we propose a dynamic scheduling approach that can deal with the emergency tasks in real time dynamically. It is actually a two-stage method to produce a temporary schedule and then to adjust the schedule as quickly as possible as is shown in Figure 2. The robust scheduling approach proposed in the paper is oriented to both dynamic emergency tasks and common tasks. We describe the flowchart of the solution in Figure 3. Before scheduling, pretreatment is conducted to compute the available opportunities for each task. Then we establish a robust satellite scheduling model

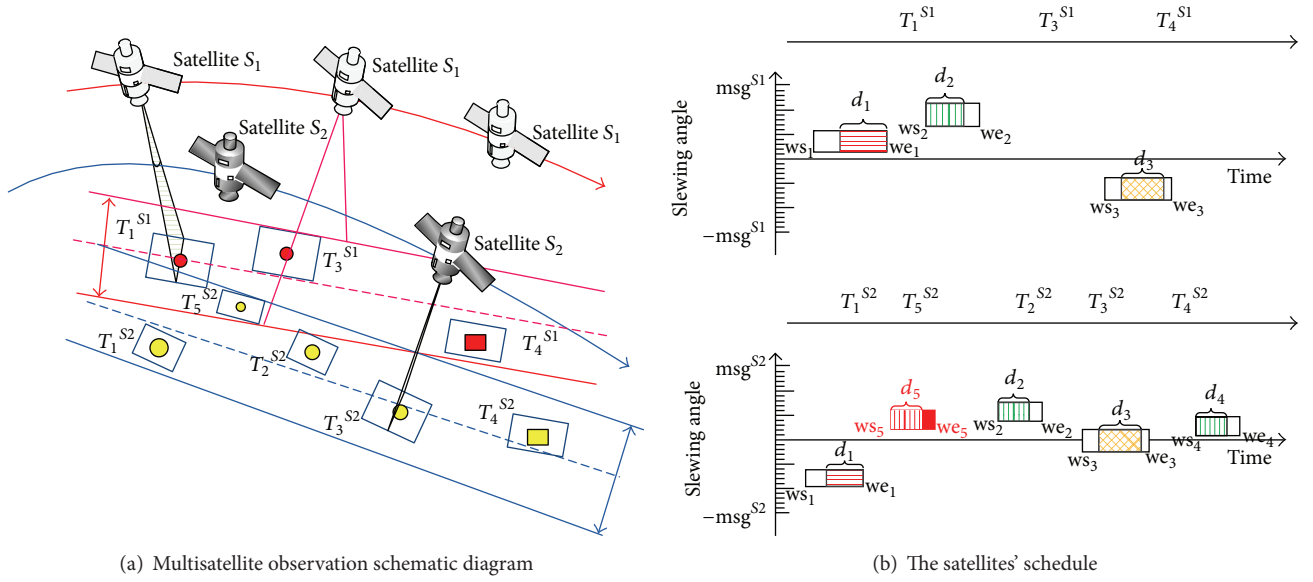


FIGURE 1: Illustration of satellite observing activity.

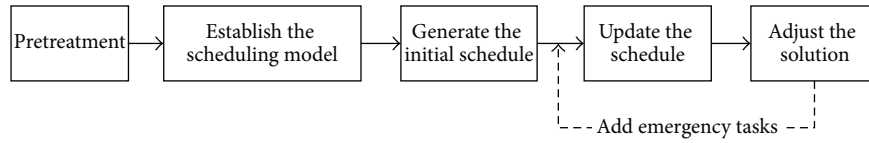


FIGURE 2: The framework of solution to the dynamic scheduling problem.

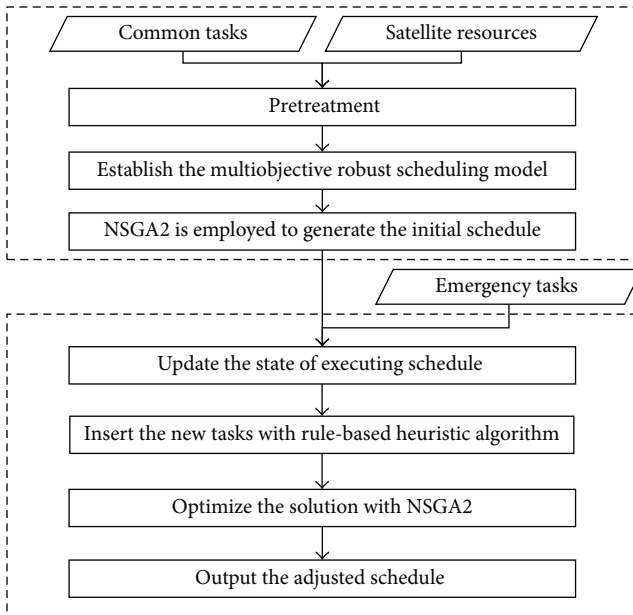


FIGURE 3: The flowchart of the dynamic scheduling approach.

with two objectives considering both revenue and robustness. To optimize objectives of the model, the multiobjective genetic algorithm NSGA2 is used to obtain robust solutions.

We can acquire an initial schedule with high revenue and stronger ability to accept emergency tasks. With the arrival of emergency tasks, we update the state of the tasks in the current executing schedule. To arrange the emergency tasks dynamically, the rule-based heuristic algorithm with compact task merging strategy is designed. Thus, we can get some temporary adjusted schedule. Lastly, NSGA2 is adopted to further improve the solution by searching the global space. It is worth mentioning that we design a neighborhood operator to initialize the population in NSGA2.

3. Model

The satellite scheduling amounts to a reasonable arrangement of satellites, sensors, time windows, and sensor slewing angle for observation tasks to maximize one or more objectives, for example, the overall observation profit, while being subject to related constraints. As a result, the satellite scheduling problem usually consists of five parts: tasks, satellite resources, opportunities, objectives, and constraints. A detailed introduction of the model is as follows.

3.1. Tasks. In this paper, we concentrate on two types of tasks: the common task and emergency task. Common imaging tasks are usually submitted to the planning system in advance. The characteristics of emergency tasks are different from those of common tasks. Emergency events usually occur

unexpectedly, with uncertainties of occurrence time and number. Thus, the emergency task number and arrival times are not known a priori. Moreover, dynamic tasks need to be completed in time, because users want to obtain observation results within certain time limits.

- (i) Common task: let $T = \{t_1, t_2, \dots, t_N\}$ be the common task set, where N is the number of common tasks. We assume that all tasks are independent, aperiodic, and nonpreemptive. Each task is associated with the weight, p_i , and the indispensable duration of task execution d_i .
- (ii) Emergency task: let $T^E = \{t_{N+1}, t_{N+2}, \dots, t_{N+N^E}\}$ be the emergency task set. Each emergency task is presented by $t_n = (p_n, d_n, at_n, dt_n)$, where p_n is the weight, d_n is the duration, at_n is the arrival time, and dt_n is the deadline.

In addition, to make the denotation of common tasks consistent with that of emergency tasks, we assume that each common task has an arrival time at $n = 0$ and a deadline $dt_n = t_s^f$.

Considering that many tasks are commonly submitted together, thus we assume in this paper that the new tasks arrive in batch style. Let $T_s = \{t_s^0, t_s^1, \dots, t_s^e, t_s^f\}$ be the scheduling time set, where t_s^i ($i \geq 1$) is the i th dynamic scheduling time, t_s^0 is the initial scheduling time, e is the total batch of emergency tasks, and t_s^f is the end time of scheduling.

Imaging tasks can be divided into two types: point target and area target. The type depends on the size of the target area and the attributes of the satellite, such as the field of view and height of satellite. The point target is small so that it can be observed by single observations trip. The area target, which covers a wide geographical area, cannot be photographed in a scene. Usually, the area target can be segmented into several point targets to be imaged. In this paper, we focus on dealing with point targets.

3.2. Satellite Resources. As for the resources, we consider a set $S = \{s^1, s^2, \dots, s^M\}$, where M is the satellite number. We assume that the sensors of the satellites considered in our study are able to slew laterally. Each resource $s^j \in S$ is denoted by $s^j = (\Delta\theta^j, \Delta d^j, sl^j, st^j, msg^j, orb^j, su^j, sd^j, duty^j)$, where $\Delta\theta^j$, Δd^j , sl^j , msg^j , orb^j , su^j , sd^j , and $duty^j$ are the field of view, the longest opening time, slewing rate, attitude stability time, maximum slewing angle, the flight time in each orbit, the start-up time of sensor, the retention time of shutdown, and the longest imaging time in each orbit.

3.3. Available Opportunities. For available opportunities, let $AO = \{ao_1^1, ao_2^1, \dots, ao_1^j, ao_2^j, \dots, ao_N^M\}$ be the opportunity set of every task on each satellite resource. For $ao_i^j \in AO$, $ao_i^j = \{ao_{i1}^j, ao_{i2}^j, ao_{i3}^j, \dots, ao_{ik_{ij}}^j\}$ represents the available opportunities of task t_i on resource s^j , where k_{ij} represents the number of available opportunities. Giving an available opportunity $ao_{ik}^j \in ao_i^j$, it is represented by $ao_{ik}^j = \{[ws_{ik}^j, we_{ik}^j], \theta_{ik}^j\}$, where ws_{ik}^j and we_{ik}^j are denoted by the start time and end time of the window W_{ik}^j , and θ_{ik}^j is the swing angle. The set of time windows of task i on satellite j is $W_i^j = \bigcup_{k \in [1, \dots, K_{ij}]} W_{ik}^j$.

3.4. Optimization Objectives. We assume that t_s^k is the current scheduling time, and then the model can be described as follows.

The primary objective is to maximize the revenue of all scheduled tasks:

$$\max : \sum_{i=1}^{N+N^E} \sum_{j=1}^M \sum_{k=1}^{K_{ij}} x_{i,k}^j p_i, \quad (1)$$

where $x_{i,k}^j$ is the decision variable whether the task i is scheduled:

$$x_{i,k}^j = \begin{cases} 1, & \text{if task } i \text{ is executed by satellite } j \text{ in the } k\text{th time window} \\ 0, & \text{otherwise.} \end{cases} \quad (2)$$

The second objective is called the neighborhood-based robust indicator [40]. The robustness based on neighborhood is proposed from the advances in robust optimization of continuous functions [41, 42]. The idea is that robust optima are located on broad peaks (plateaus) in the fitness landscape, whereas brittle optima are located on narrow peaks, as shown in Figure 4. A tradeoff is made between the broadness and height of the peaks, such that a low and broad peak may be preferable to a narrow and high peak. The problem of satellite scheduling is a discrete problem. There are some schedules which are different sequence of the scheduled tasks ordered in time but have the same revenue. If only

considering the revenue, any one of these schedules is a feasible solution. However, dynamic task scheduling not only requires arranging more emergency tasks, but also requires reducing the frequency and difficulty of adjustment as much as possible. Thus, the property of resisting disturbances and self-adjusting is very important for dynamic task scheduling.

Hence, we define the neighborhood-based robustness as the total revenue of scheduled tasks that can be reassigned to other timeslots in the scheme directly. It is used to measure the ability of a scheduling scheme to rearrange scheduled tasks. The higher value of the neighborhood-based robust indicator is, the more possible it is to rearrange scheduled

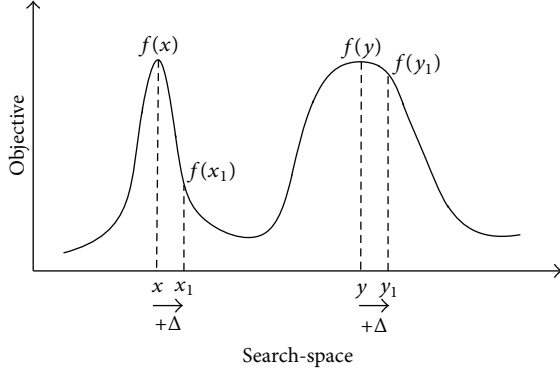


FIGURE 4: The idea in the robust optimization technique for continuous problem [47].

tasks. Therefore, we maximize the value of the neighborhood-based robust indicator:

$$\max : \sum_{i=1}^{N+N^E} \sum_{j=1}^M \sum_{k=1}^{K_{ij}} p_i \times x_{i,k}^j \times y_i, \quad (3)$$

where y_i is the parameter that indicates whether the task $t_i \in$ SS can be rearranged in another timeslot:

$$y_i = \begin{cases} 1, & \text{if task } t_i \text{ can be rearranged in another timeslot} \\ 0, & \text{otherwise.} \end{cases} \quad (4)$$

3.5. Constraints. There are some constraints that are used to analyze the complex restricted condition in the model.

(1) Each task t_i must be executed in an available opportunity $ao_{ik}^j, ao_{ik}^j \in ao_i^j$.

Hence, we have the available opportunity constraint:

$$C_1 : \begin{cases} x_{i,k}^j (ts_i^j - ws_{ik}^j) \geq 0 \\ x_{i,k}^j (ts_i^j + d_i - we_{ik}^j) \leq 0 \\ a_i \leq t_s^k \leq ts_i^j \leq dt_i, \end{cases} \quad (5)$$

where ts_i^j denotes the start time of task t_i . Actually, for a given scheduled task t_i on s^j , its starting observation time ts_i^j and observation angle θ_i^j are determined. All scheduled tasks form the current schedule $SS = \bigcup_{j \in [1, M]} T^j$, where T^j is a sequence of scheduled tasks on satellite j .

(2) Each task only needs to be executed once, and the execution process does not involve preemptive service. So we have the following uniqueness constraint:

$$C_2 : \sum_{j=1}^M \sum_{k=1}^{K_{ij}} x_{i,k}^j \leq 1. \quad (6)$$

(3) There should be an adequate transmission time for sensor in any two adjacent tasks t_u, t_v assigned to the same satellite s^j to finish the actions such as shutting down,

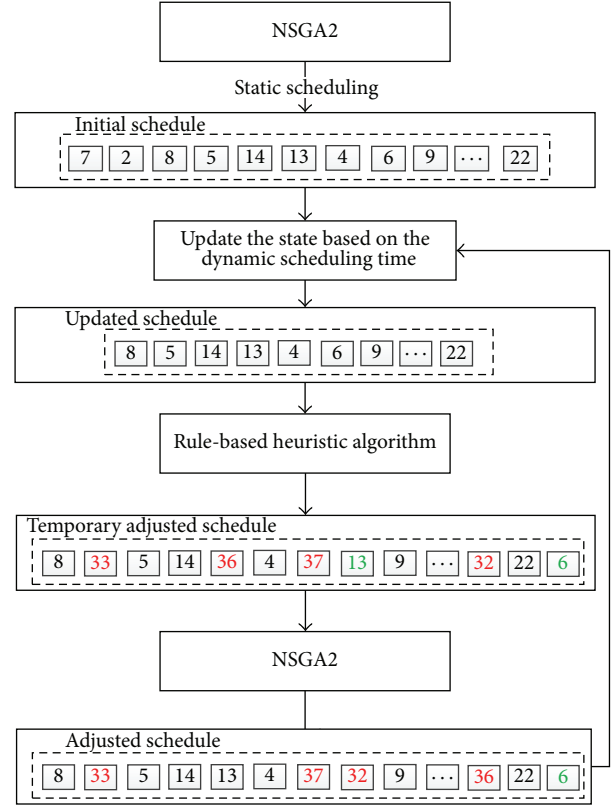


FIGURE 5: The flowchart of the algorithm.

swinging the observation angle, stabilizing gesture, and start-up. So we have the switch time constraint:

$$C_3 : te_u^j + sd^j + sl^j \times |\theta_v^j - \theta_u^j| + st^j + su^j \leq ts_v^j. \quad (7)$$

(4) Another constraint is the total imaging time of any satellite, which should be less than the allowable longest imaging time of its sensor during every time period. Let the scheduling period be $[t_s^k, f]$. Hereby, we have the imaging time constraint:

$$C_4 : \sum_{i \in T_{t_b}^j} d_i \leq \text{duty}^j, \quad (8)$$

where $T_{t_b}^j$ denotes a sequence of scheduled tasks on satellite j which flies during the time span $[t_b, t_b + \text{orb}^j]$, where $t_b \in [t_s^k, f - \text{orb}^j]$.

4. Algorithms

In this section, we introduce the dynamic emergency tasks scheduling algorithm HA-NSGA2 proposed in this paper. The algorithm, which is actually a combined algorithm, aims at improving the efficiency of dynamic emergency tasks scheduling. The flowchart of the algorithm is depicted in Figure 5. Firstly, the multiobjective genetic algorithm NSGA2 with two objectives, that is, revenue and robustness, is used for common tasks scheduling to obtain the optimal solution.

With the arrival of emergency tasks, the rule-based heuristic algorithm is applied to insert these emergency tasks into the initial schedule as much as possible. To improve the imaging efficiency as much as possible, we propose a compact task merging method that considers task execution duration in the rule-based heuristic algorithm. If two or more targets are geographically adjacent, they may be combined into a compact composite task by the compact task merging method. The details of compact task merging strategy are described in Appendix B. Based on the rule-based heuristic algorithm, the multiobjective genetic algorithm NSGA2 is employed to find the optimal solution. The neighborhood operator, which is embedded in NSGA2, aims at generating the initial population.

4.1. Multiobjective Genetic Algorithm. Satellites observation scheduling problem has been proven to be NP-complete. Thus, it is impossible to find polynomial time algorithms for finding optimal solutions. Genetic algorithms are the most successful algorithms for efficiently dealing with the complexity of computationally hard problems from the networking domain [23, 43]. Before introducing the multiobjective genetic algorithm NSGA2, we firstly make clear some terms, such as Pareto Dominance, Pareto optimality, Pareto Optimal Set, Pareto Front, Nondominated Sorting, and crowding distance, which will be referred to in the algorithm. We give the associative definitions in Appendix A.

As shown in Figure 6, we present the main process of NSGA2. First, we initialize the parameters and generate the initial population P_{father} . After computing the fitness function of P_{father} , we make use of the binary tournament selection mechanism, single point crossover operator, and single point mutation operator for the current individuals until the number of the next generation population P_{child} reaches the size M . Then, we set the total population as $Q = P_{\text{father}} \cup P_{\text{child}}$ by combining P_{father} and P_{child} . Next, the nondominated sorting algorithm is used to divide the population Q into different Pareto Front F_{rank} with the rank of each individual. We choose the individuals from F_{rank} until the number of the next population $|P_{\text{next}}|$ satisfies the request, ranking from small to large. If $|P_{\text{next}}|$ is greater than M , then we sort the current F_{rank} in order according to the crowding distance and choose the better individuals as P_{next} . Finally, if the result satisfies the termination conditions, then we output the nondominated solution set. Otherwise, we let $P_{\text{father}} = P_{\text{next}}$ and repeat the process.

The main parts of the genetic algorithm will be described in detail including initialization, selection operator, crossover operator, mutation operator, nondominated sorting, and the computation of crowding distance.

4.1.1. Initialization. Unlike local search techniques, the genetic algorithm uses a population of individuals giving thus the search a larger scope and chances to find a better solution. In this paper, the satellite scheduling problem is encoded into a set of chromosomes. Each chromosome composed of an imaging task sequence of different satellites represents a solution and is assigned a fitness value to determine how good each chromosome is. The structure of a chromosome

is shown in Figure 7. In the algorithm, some parameters such as the size of population M , the maximum iterative number T , the probability of crossover $p_{\text{crossover}}$, and mutation probability p_{mutation} must be set. The initial population is generated by random strategy.

4.1.2. Selection Operator. Several selection strategies have been proposed in the previous literatures such as roulette wheel selection and binary tournament selection. We adopt binary tournament selection in this paper. The main steps are to choose two individuals randomly from a population and to select the individual that dominates the others.

4.1.3. Crossover Operator. Crossover operator plays an important role in the genetic algorithm. The best genetic information of parents can be transmitted to offspring during generations of the evolution process by the crossover operator. After parent chromosomes have been selected from the population, we set the crossover probability $p_{\text{crossover}}$ to allow the majority of chromosomes to mate and pass to their offspring. Figure 8 shows an example of the crossover. According to the characteristics of the satellite scheduling problem, we cross the parent chromosomes to generate child chromosomes and then delete the same genes in common chromosome. Finally we attempt to arrange the tasks differently for both parent chromosomes in child chromosomes.

4.1.4. Mutation Operator. Mutation operator intends to keep the variety of the population and improve the individuals by small local perturbations. Additionally, the mutation operator provides a component of randomness in the neighborhood of the individuals of the population. Therefore, genetic algorithm has the ability to avoid falling prematurely into local optima and eventually escapes from them during the search process. For example, if we set the probability $p_{\text{mutation}} = 0.2$, then the overall procedure of the random mutation operator on the chromosome is shown in Figure 9. We choose a scheduled task that has another window randomly such as t_8 . Then the task is rearranged in another available window.

4.1.5. Nondominated Sorting. The main procedure of nondominated sorting is described in Algorithm 1 in detail.

4.1.6. Crowding Distance Computation. The main procedure of computing the crowding distance is described in Algorithm 2 in detail.

4.2. Rule-Based Heuristic Algorithm. The solution with maximum revenue and robustness, which is produced by NSGA2, is selected as the initial schedule. When the schedule is executed, a number of emergency tasks arrive. In this case, the current executing schedule must be modified to arrange these emergency tasks.

When a batch of emergency tasks arrives, based on the current scheduling time, tasks may have different states: finished task, running task, waiting task, and new task. In the example shown in Figure 10, t_s^k is the current scheduling time. If task i has been executed before the current time, then task

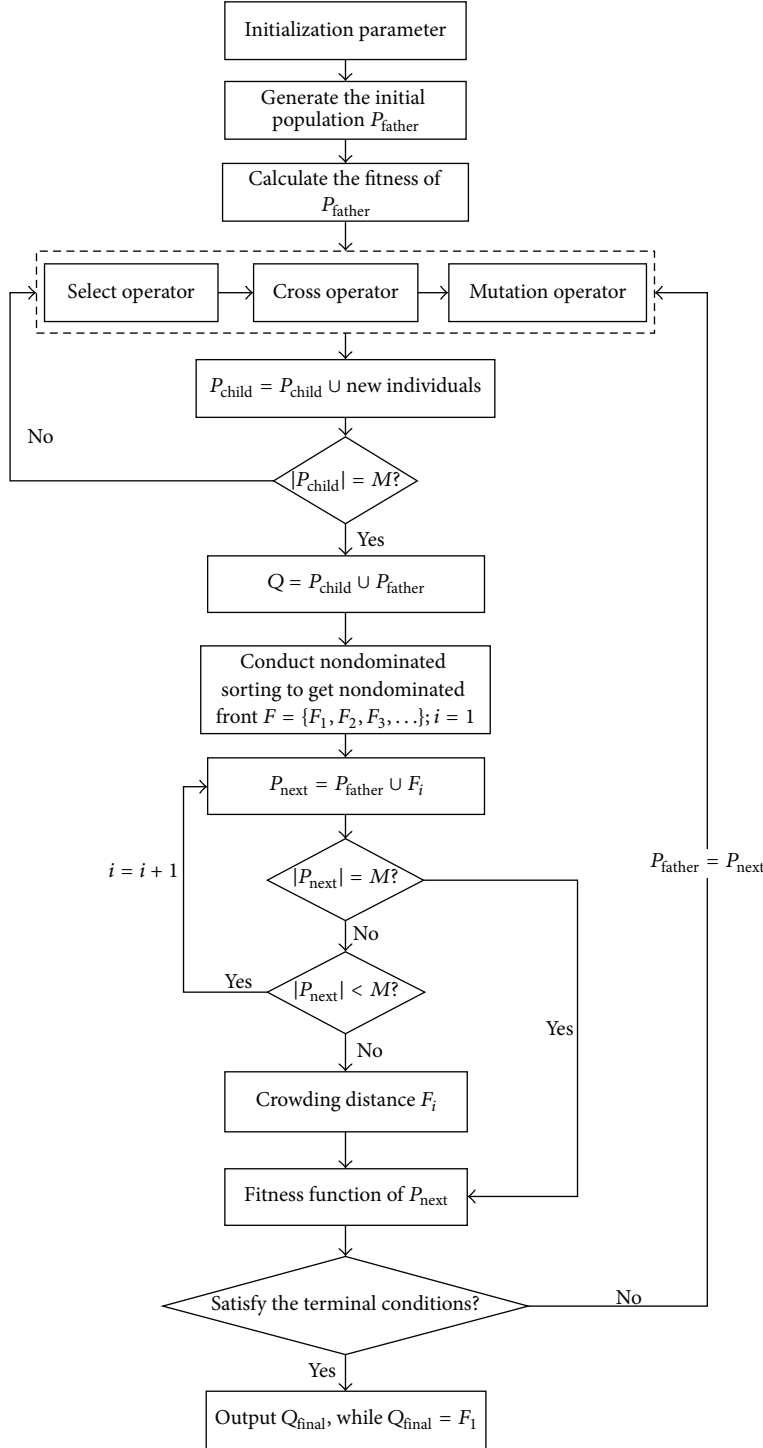


FIGURE 6: The main process of NSGA-2.

i is a finished task while $te_i < t_S^k$. If $ts_i < t_S^k < te_i$, then task i is a running task. Task i is a waiting task if $ts_i > t_S^k$. If task i is an emergent task that has not been planned, then task i is a new task.

According to the analysis on the state of the tasks, the waiting tasks in the current schedule can be adjusted. Focusing on the new emergency tasks, we assign these

tasks to the current schedule. In this paper, we employ a rule-based heuristic algorithm considering compact task merging (CTM-DAHA) to insert the new tasks dynamically [44]. The main idea of the heuristic algorithm is to assign the most conflict-free time window to the task most in need. With CTM-DAHA, we obtain the temporary adjusted schedules.

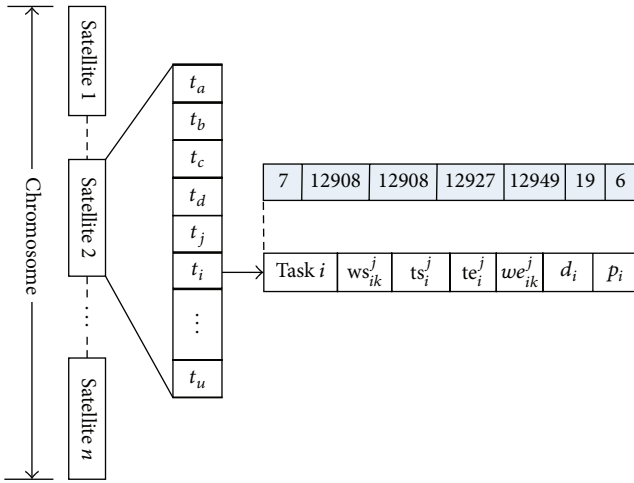


FIGURE 7: The structure of chromosome.

To describe the dynamic adjusting rule-based heuristic algorithm considering compact task merging clearly, we first give some definitions.

Definition 1. The need of the task indicates how badly the task needs to be performed [30, 44]:

$$N(t_i) = \frac{p_i}{|AO_i|}. \quad (9)$$

Definition 2. The contention of the task window is measured by counting the number of tasks that need the time window, weighted by the weights of the tasks:

$$C(w) = \sum_{t_i \in T(w)} p(t_i), \quad (10)$$

where $T(w)$ is the set of tasks that could be executed at any moment within time window w and $p(t_i)$ is the weight of the tasks t_i .

The dynamic adjusting rule-based heuristic algorithm CTM-DAHA is presented as shown in Figure 11 [44]. To accommodate emergency tasks in the initial scheme, firstly we update the state of tasks in the schedule. Then, all new tasks are inserted into a waiting queue according to the needs of the tasks $N(t_i)$, from high to low. The time windows of each task are ranked from low to high according to the contention $C(w)$. For a task, we first judge whether it can merge with any other existing task in the current schedule. If all time windows of the task fail to be merged, we attempt to insert the task directly if it does not conflict with any other task. If direct insertion fails, we judge whether the task can be inserted by rearranging the conflict tasks. If direct insertion and insertion by rearranging do not succeed, some tasks may be deleted for inserting an emergency task. Then the entire process will repeat until the waiting queue is empty. Take, for example, a current executing schedule as shown in Figure 12(a). With the arrival of emergency tasks, that is, $T^E = \{t_8, t_9, t_{10}, t_{11}\}$, they should be arranged in time. Task t_8 can be combined

with t_2 by compact task merging. Task t_9 can be inserted directly. Task t_{10} is inserted into the schedule by rearranging the conflict task t_5 . Task t_7 is deleted from the schedule so that task t_{11} with higher weight can be inserted. Figure 12(b) shows the planning task list of the adjusted schedule.

4.3. Neighborhood Operator. The solution obtained by CTM-DAHA still needs to be optimized. Based on the temporary adjusted schedule, we use NSGA2 to find the optimal solution. Different from the multiobjective genetic algorithm described in Section 4.1, we design a neighborhood operator to generate the initial population.

The main idea of the neighborhood operator is to search all neighborhoods of the current schedule. As shown in Figure 13, the current schedule SS is represented by the red point and the yellow points denote the neighborhoods of SS. Specifically, we change the locus of a task to move it to another position, as shown in Figure 14.

The procedure of the neighborhood operator is described as shown in Algorithm 3.

After initializing the population using the neighborhood operator, the next serial of operation is selection, crossover, mutation, and nondominated sorting as expressed in Section 4.1. Finally, using the multiobjective genetic algorithm NSGA2, we obtain the optimal solution as the new adjusted schedule.

5. Experimental Simulation and Discussion

In this section, we evaluate the performance of the proposed algorithm HA-NSGA2. To reveal the advantages of the algorithm, we compare it with RBHA-DTSP [36] and CTM-DAHA [44]. The two methods both use multiobjective genetic algorithm to make the initial schedule for common tasks. To evaluate the advantage of the compact task merging method (CTM), we perform experiments to compare the scheduling results obtained by HA-NSGA2 with the two task merging strategies: traditional task merging strategy (TTM) and compact task merging method (CTM). Specifically, we embed the two task merging methods into HA-NSGA2, respectively, so as to prove the effectiveness of compact task merging strategy.

- (i) RBHA-DTSP: the basic idea of RBHA-DTSP is to carry out some amount of iterative repair search inside each emergency task's available opportunities. With the repair search for a given emergency task, some rules are adopted to decide which tasks should be retracted [36].
- (ii) CTM-DAHA: this is a dynamic adjusting rule-based heuristic algorithm considering compact task merging (CTM-DAHA) which is designed to insert the emergency tasks into the initial schedule.

To evaluate the result we use the following performance metrics.

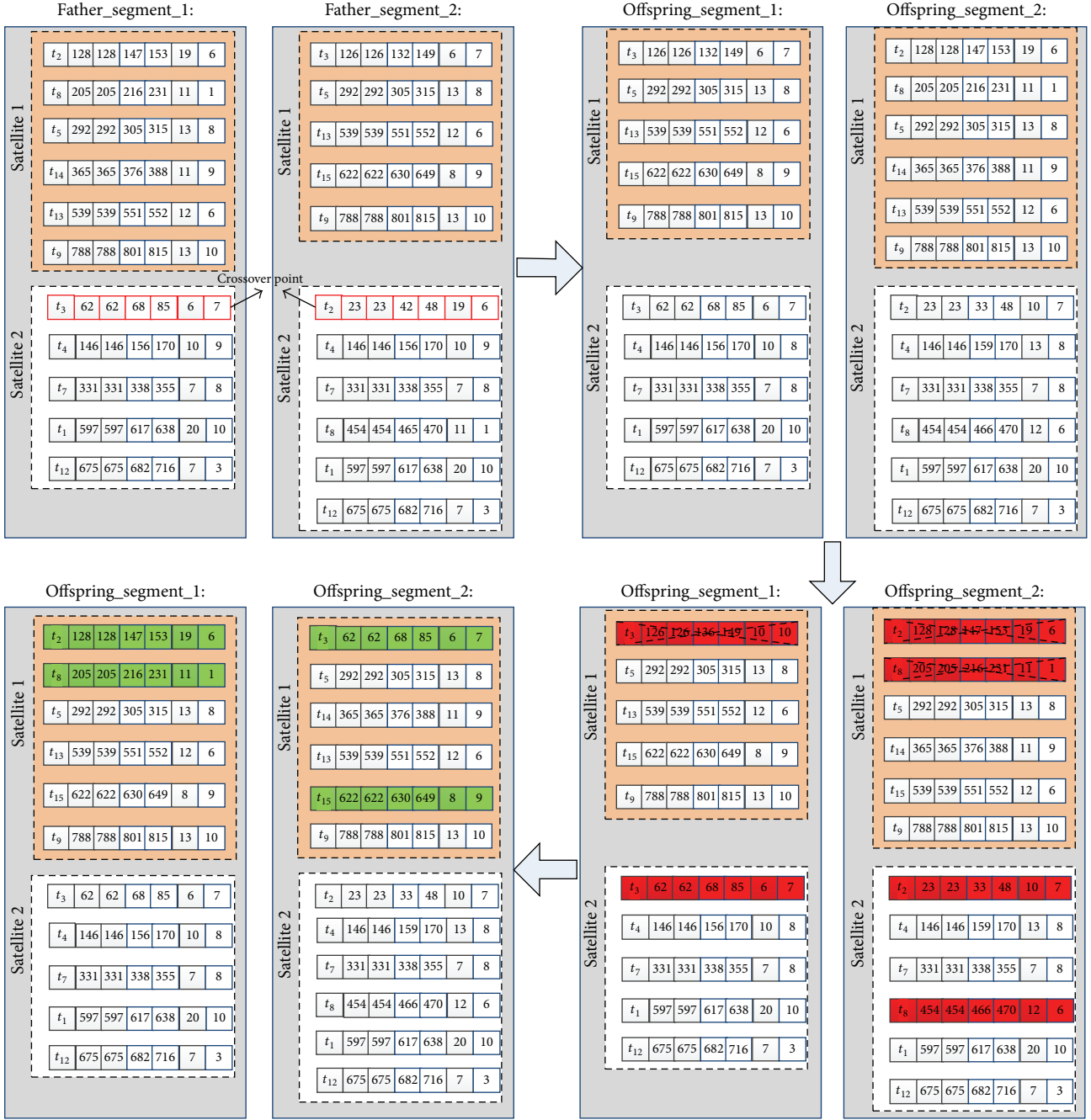


FIGURE 8: Crossover operator.

Emergency task revenue (ETR) can be described as

$$\text{ETR} = \sum_{i=1}^N \sum_{j=1}^M \sum_{k=1}^{K_{ij}} x_{i,k}^j p_i. \quad (11)$$

Satisfaction ratio (SR) is defined as SR = total number of tasks that can be arranged in the scheme/total number of tasks.

Scheduled tasks robustness (STR) can be defined as

$$\text{STR} = \sum_{i=1}^N \sum_{j=1}^M \sum_{k=1}^{K_{ij}} p_i \times x_{i,k}^j \times y_i. \quad (12)$$

In fact, the satellite application is a complex process and the orders can only be uploaded with special equipment within limited visible time windows, so it is important to make the least adjustment to keep the stability of the schedule as well as to insert more tasks to keep the efficiency of the

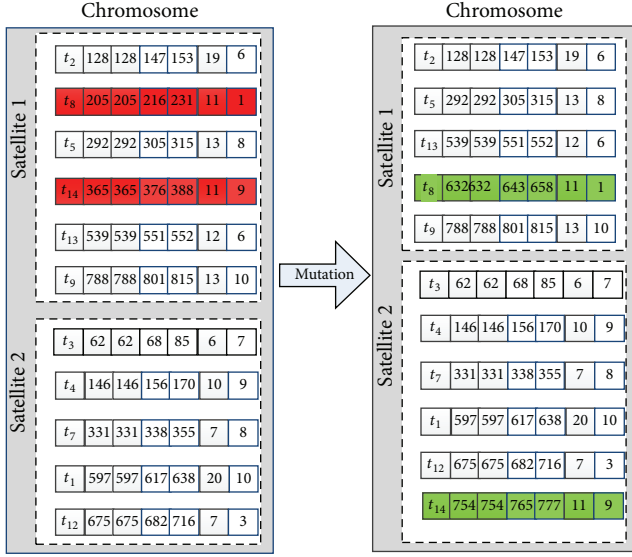


FIGURE 9: Mutation operator.

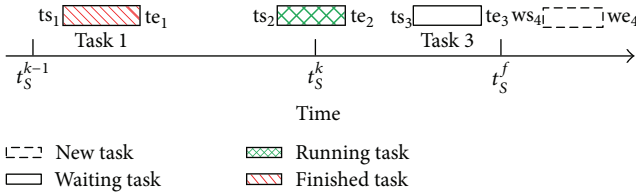


FIGURE 10: The states of tasks.

schedule. Therefore, the main measurement of perturbation needs to be considered.

Perturbation measurement (PM) is a metric used to describe the distance between the adjusted scheme and the initial scheme. PM can be defined with the total number of initial tasks that change in the adjusted scheme compared with the initial scheme.

5.1. Simulation Method and Parameters. Without loss of generality, we acquire the imaging targets randomly in the area: latitude $-180^\circ \sim 180^\circ$ and longitude $-65^\circ \sim 65^\circ$. The scheduling period is 24h, which can be denoted by $[0, 86400]$. The number of common tasks is 200. In addition, there are five batches of new coming emergency tasks and each batch is given the same size: 50. The arrival time of a batch t_s^k is a random positive number distributed in $[t_s^0, t_s^f]$. In the experiment, we simulate 5 batches of emergency tasks and T_s is set as $T_s = \{t_s^0, t_s^1, \dots, t_s^e, t_s^f\} = \{0, 59, 13300, 26100, 41800, 65000, 86400\}$. Moreover, the deadline of an emergency task is the end time of scheduling: 86400. Without loss of generality, the weights of all tasks are randomly distributed in $[1, 10]$. Two sensors on different satellites are simulated to accomplish the tasks. The parameters of sensors are shown in Table 1. The available opportunities, including time windows and slewing angles associated with tasks, are computed using STK (Satellite

TABLE 1: Parameters of satellites.

Parameters	Satellite s^1	Satellite s^2
Average height (km)	780.099	505.984
Orbital inclination (degree)	98.5	97.421
Local descending node time (am)	10:30	10:30
Argument of perigee	90	90
Eccentricity ratio (degree)	0.0011	0
Nodical period (min)	100.38	97.716
Maximum slewing angle (degree)	± 32	± 32
Field of view (degree)	8.4	6

Tool Kit). The time window is removed if its span is shorter than the duration of the corresponding task or its finished time is later than the deadline. Moreover, we assume that the maximum time for each satellite to open its sensor once is 60 seconds and that the longest imaging time in any period time of orb^j is 150 seconds.

5.2. The Simulation Result. We carry out two groups of experiments in this section. We investigate the performance impact of different algorithms, that is, RBHA-DTSP, CTM-DAHA, and HA-NSGA2, in the first group. The experimental results are depicted in Figure 15. The second group of experiments is designed to evaluate the efficiency of the task merging method: CTM and TTM.

Figure 15(a) shows the result of the revenue of scheduled emergency tasks in each adjusted schedule. Among the algorithms, HA-NSGA2 shows the best scheduling revenue. The explanation is that HA-NSGA2 optimizes the robustness and uses the compact task merging strategy. In contrast, CTM-DAHA does not improve the robustness and RBHA-DTSP does not consider task merging. Based on Figure 15(b), we can see that, with more batches of emergency tasks, the robustness of each adjusted schedule decreases. This can be attributed to the fact that, with the time of scheduling increasing, the scheduling period becomes shorter. This means that the tasks' available opportunities in a schedule decrease and the number of tasks which can be rearranged decreases. As a result, the robustness becomes smaller. In addition, in each scheduling, HA-NSGA2 can generate a more robust schedule except the last scheduling. In the last scheduling, the schedule obtained by HA-NSGA2 shows the lowest robustness because HA-NSGA2 scheduled more tasks and less spare time is left for other tasks to be rearranged. As a result, the robustness is worse. Figure 15(c) presents the satisfaction ratio of every scheduling result. It is found that the satisfaction ratio of all algorithms decreases as the arrival of more emergency tasks. This can be attributed to the fact that the number of scheduled tasks decreases compared with the increasing emergency tasks. Moreover, HA-NSGA2 shows the highest satisfaction ratio among all of the algorithms. Table 2 describes the number of scheduled emergency tasks and the number of adjusted tasks in the adjusted schedules. It is found that there is little difference of disturbance among the algorithms. On the whole, the ability

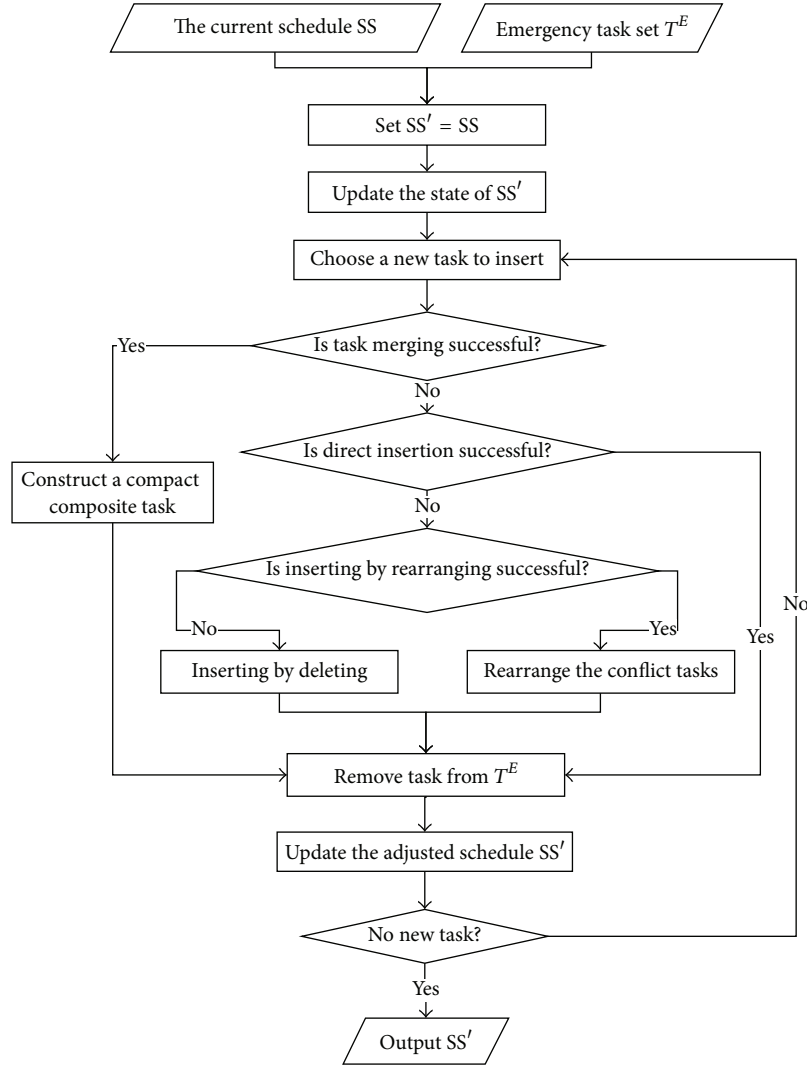


FIGURE 11: The process of inserting a new task.

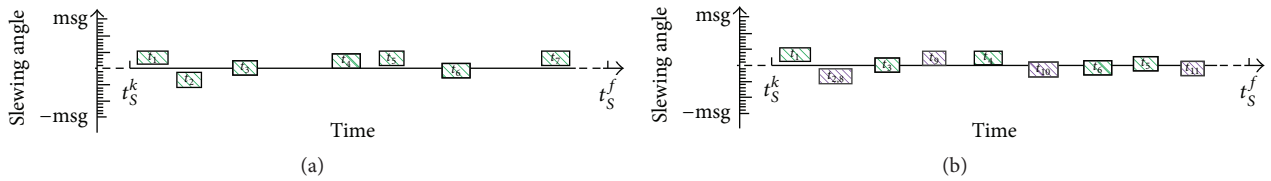


FIGURE 12: (a) The current executing schedule. (b) The adjusted schedule.

dealing with emergency tasks of HA-NSGA2 is superior to that of the other algorithms.

Figure 16 shows the performance impact of the two task merging methods. Figure 16(a) demonstrates that, with the increase in scheduling time, the compact task merging method (CTM) can get a higher total emergency task revenue. This result indicates that more emergency tasks can be arranged by the compact task merging method and more opportunities can be left for other emergency tasks. From Figure 16(b), it is found that robustness of the schedule of the compact task merging method is lower than that

of the traditional task merging strategy. This result can be attributed to two factors. First, the algorithm with the compact task merging strategy can arrange more emergency tasks compared with traditional task merging method. The other is the composite task, which is unable to participate in the robustness computing because there is only one synthetic available window, acquired by compact task merging. Moreover, the number of composite tasks by compact task merging is more than that by traditional task merging. So the compact task merging strategy is better than the traditional merging strategy in the available windows and the satellite

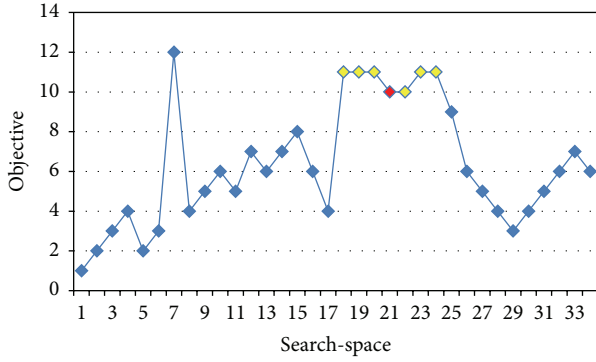


FIGURE 13: The neighborhoods of the current schedule.

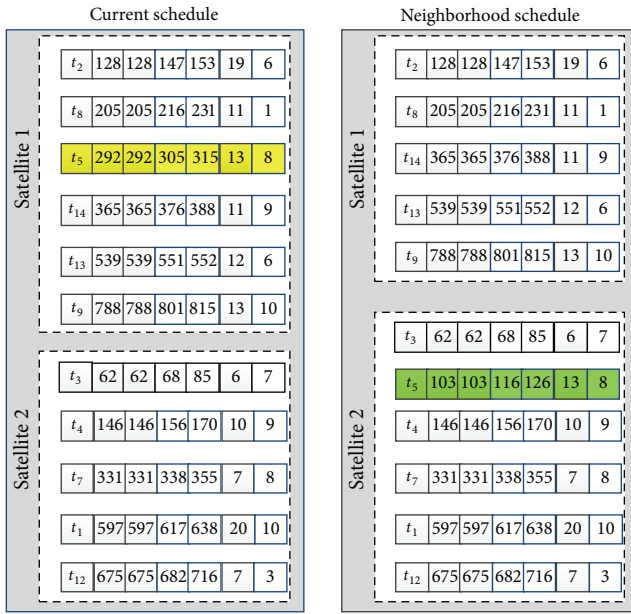


FIGURE 14: Illustration of neighborhood operator.

TABLE 2: The perturbation measurements of different algorithms.

Batch	RBHA-DTSP		CTM-DAHA		HA-NSGA2	
	N	PM	N	PM	N	PM
1	47	2	47	2	48	4
2	46	5	46	5	47	5
3	41	3	41	1	42	1
4	43	3	46	4	47	4
5	43	1	42	1	44	1

Note: N: the number of scheduled emergency tasks.

resources left. Figure 16(c) depicts that the compact task merging method can arrange more emergencies and obtain a more satisfactory schedule. Table 3 describes the way of adjusting of emergency tasks and the distance of the adjusted scheme from the initial scheme. In the same perturbation measurement, HA-NSGA2 with the compact task merging method can arrange more emergency tasks. Also, CTM can merge more tasks than CTM. The result proves that

TABLE 3: The perturbation measurements of different task merging strategies.

Batch	Method	N	TM	NI	NIR	NID	PM
1	CTM	48	4	42	1	1	4
	TTM	47	1	44	1	1	4
2	CTM	47	4	38	3	2	5
	TTM	46	1	41	2	2	6
3	CTM	42	3	38	1	0	1
	TTM	41	2	38	1	0	2
4	CTM	47	7	36	1	3	4
	TTM	46	1	39	1	4	6
5	CTM	44	3	40	0	1	1
	TTM	42	1	43	0	0	1

Note: N: the number of scheduled emergency tasks, PM: the number of adjusted tasks in the schedules, NITM: the number of emergency tasks inserted by task merging, NI: the number of emergency tasks inserted directly, NIR: the number of emergency tasks inserted by rearranging, and NID: the number of emergency tasks inserted by deleting.

the compact task merging strategy has an advantage over the existing traditional method. The compact task merging method can improve the chances of inserting emergency tasks.

5.3. Discussion. Based on the extensive experiments, we can evaluate the performance of HA-NSGA2 proposed in the paper. Using the algorithm HA-NSGA2 with two objectives, that is, the revenue and robustness, we can obtain the scheduling scheme that has higher revenue and stronger ability to resist disturbance in emergency tasks scheduling. HA-NSGA2 algorithm can insert more emergency tasks while keeping the high revenue as is shown in Figure 15. Moreover, HA-NSGA2 can significantly reduce the scheduling disturbance in the process of scheduling emergency tasks. The experiments prove that the HA-NSGA2 is capable of finding optimal solutions with the ability to resist disturbance and to address emergency tasks efficiently. Moreover, the compact task merging strategy can improve the chances of merging multiple tasks and save the satellite resources. So the method proposed in this paper is more practical and satisfies the requests of the users.

6. Conclusion and Future Work

To solve dynamic scheduling problem for emergency tasks in a multisatellite observation system, we propose a mixed algorithm named HA-NSGA2 to improve the ability of emergency tasks scheduling from two aspects. In our algorithm two optimization objectives, that is, revenue and robustness, are considered continuously. Firstly, we schedule common tasks by using the multiobjective genetic algorithm NSGA2 to obtain an initial solution with high revenue and robustness. Then, the heuristic algorithm which embeds a compact task merging strategy is used to handle emergency tasks. Finally, to optimize the solution, NAGA2, which embeds a novel neighborhood operator to generate the initial population,

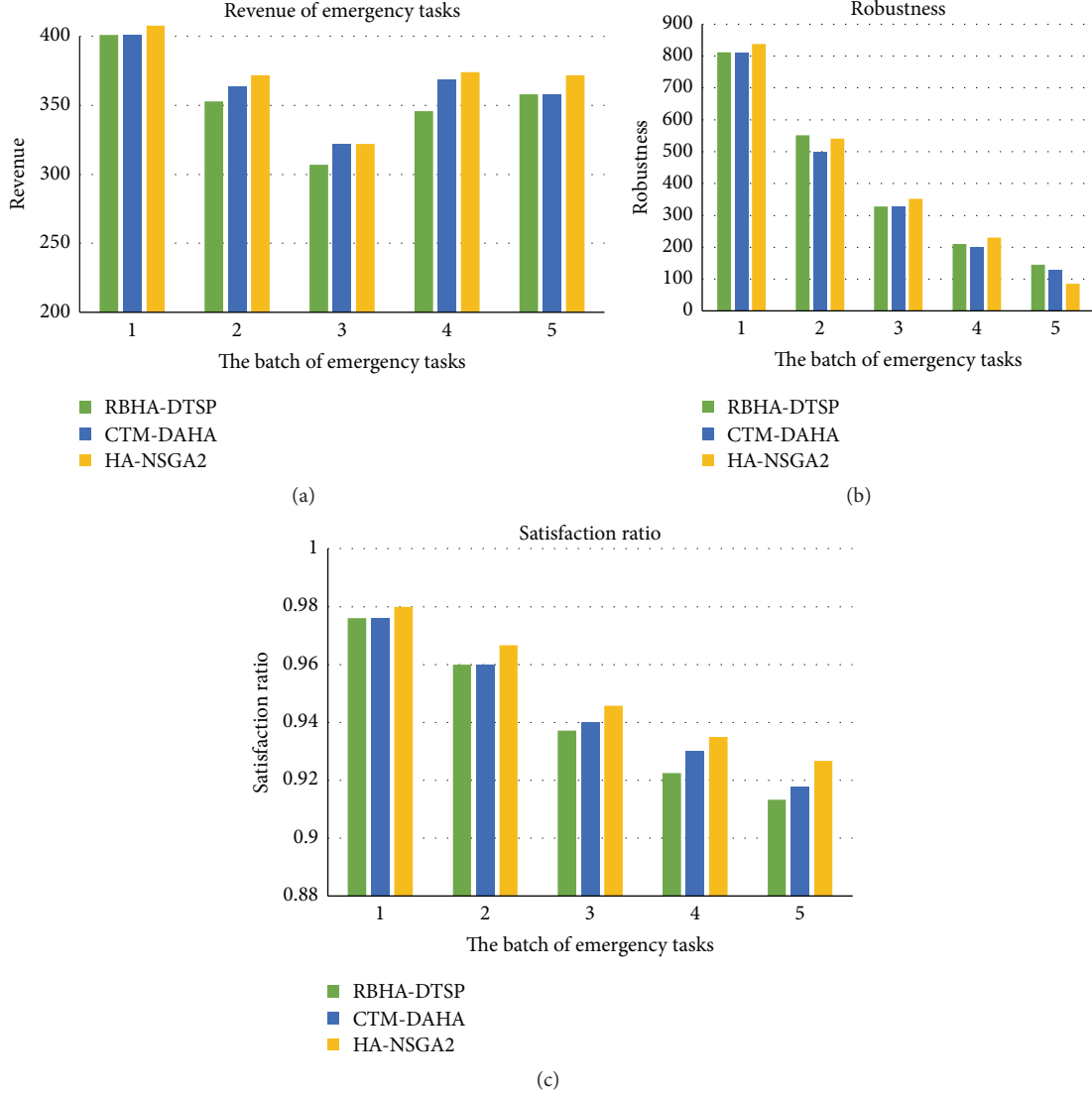


FIGURE 15: The performance of different dynamic scheduling algorithms.

is adopted to search the optimal result in the local scope. Extensive experimental simulation proves that the robustness is an important factor in dynamic scheduling. The algorithm HA-NSGA2 proposed in this paper optimizes the robustness and obtains the more robust scheduling results with the highest revenue compared with other algorithms. Moreover, the compact task merging strategy can improve the chances of merging multiple tasks. Therefore, we conclude that the robustness of the schedule has considerable significance in the dynamic scheduling problem.

In the future, our studies will address the following issues: first, we will conduct some research on area targets and consider more critical issues, such as resource availability, data dimension, and communication. In addition, we will improve our scheduling model by considering more constraints, such as memory capacity, energy consumption, and speed of upload. Finally, we will extend our experiments in real scene.

Appendices

To help the readers understand, we make some terms clear that will be referred to in NSGA2 and describe the compact task merging strategy in detail in these appendices. In Appendix A we introduce some definitions of the multiobjective genetic algorithm NSGA2. Compact task merging strategy is presented in Appendix B.

A. The Associative Definitions for NSGA2

Some fundamental definitions of the multiobjective genetic algorithm NSGA2 are given in the following [44, 45].

Definition for Pareto Dominance. For any $\mathbf{u} \in \mathbb{R}^m$ and $\mathbf{v} \in \mathbb{R}^m$: \mathbf{u} dominates \mathbf{v} (in symbols $\mathbf{u} < \mathbf{v}$) if and only if: $\forall i = 1, \dots, m$: $\mathbf{u} \leq \mathbf{v}$ and $\exists i \in \{1, \dots, m\}$: $\mathbf{u} < \mathbf{v}$.

```

for each  $p \in Q$  // every individual in population  $p \in Q$ 
     $S_p = \emptyset$ ; // let the set of individuals dominated by  $p$  be null
     $n_p = 0$ ; // the number of individuals dominating  $p$  be 0
    for each  $q \in Q$ 
        if ( $p < q$ ) then // if  $p$  dominates  $q$ 
             $S_p = S_p \cup q$ ; add  $q$  into  $S_p$ 
        else if ( $p > q$ ) then // if  $q$  dominates  $p$ 
             $n_p = n_p + 1$ ;
    end for;
end for;
for each  $p \in Q$ 
    if ( $n_p == 0$ ) then
         $p_{\text{rank}} = 1$ ; // the rank of  $p$  is 1st
         $F_1 = F_1 \cup p$ ; // add  $p$  into the set of the first non-dominated front  $F_1$ 
    end for;
 $i = 1$ ; // initialize the rank of the non-dominated fronts
while ( $F_i \neq \emptyset$ )
    for each  $p \in F_i$ 
        for each  $q \in S_p$ 
             $n_q = n_q - 1$ ; // the number of individuals dominating  $p$  is reduced by one
            if ( $n_q == 0$ ) then // if there is no individual dominating  $q$ 
                 $q_{\text{rank}} = i + 1$ ; // the rank of  $q$  plus one
                 $F_{i+1} = F_{i+1} \cup q$ ; // add  $q$  into the non-dominated front  $F_{i+1}$ 
            end for;
        end for;
    end for;
     $i = i + 1$ ;
end while.

```

ALGORITHM 1

```

 $I = |F_{\text{rank}}|$ ; // number of solutions in  $F_{\text{rank}}$ 
for each  $p \in F_{\text{rank}}$ 
     $F_{\text{rank}}[p]_{\text{distance}} = 0$ ; // initialize the distance
end for;
for each objective  $m$ 
     $F_{\text{rank},m} = \text{sort}(F_{\text{rank}}, m)$ ; // sort using each objective value
    for each  $i = 2 \rightarrow I - 1$ 
         $F_{\text{rank},m}[i]_{\text{distance}} = (F_{\text{rank},m}[i + 1] - F_{\text{rank},m}[i - 1]) / (F_{\text{rank},m}^{\text{max}} - F_{\text{rank},m}^{\text{min}})$ ; //crowding-distance of individual  $i$  in  $m$ _th objective
    end for;
end for;
for each  $p \in F_{\text{rank}}$ 
    for each objective  $m$ 
         $F_{\text{rank}}[p]_{\text{distance}} = F_{\text{rank}}[p]_{\text{distance}} + F_{\text{rank},m}[p]_{\text{distance}}$ ; // crowding-distance of  $p$ 
    end for;
end for.

```

ALGORITHM 2

```

(1) According to the arrival time of a batch of emergency tasks, select the waiting tasks in the current schedule SS as  $T^w$ 
(2) for all  $i \in [1, 2, \dots, |T^w|]$ , do
(3) for all  $j \in [1, \dots, M]$ , do
(4) compute all available opportunities for task  $i$  on satellite  $j$ 
(5) end for
(6) compute the number of available opportunities  $|W_i|$  for task  $i$ 
(7) end for
(8) let the neighborhood set of the current schedule  $SS^N$  be  $SS^N = SS$ 
(9) while  $T^w \neq \emptyset$  do
(10) take the serial number  $i$  of the first task in  $T^w$ 
(11) if  $|W_i| \leq 1$  do
(12) remove  $t_i$  from  $T^w$  and then go to Step (9)
(13) end if
(14) else
(15) try to change the locus of task  $i$  in the chromosome
(16) if succeed, update the chromosome and add the chromosome to  $SS^N$ 
(17) remove  $t_i$  from  $T^w$  and then go to Step (9)
(18) end if
(19) else
(20) remove  $t_i$  from  $T^w$  and then go to Step (9)
(21) end while
(22) output the neighborhood set of the current schedule  $SS^N$ 

```

ALGORITHM 3

Definition for Pareto Optimality. The idea that a candidate solution $\mathbf{x} \in \Omega$ is the optimal solution of Pareto means that $\mathbf{x} \in \Omega$ makes $F(\mathbf{x}') < F(\mathbf{x})$. \mathbf{x} is a K -dimensional decision variable in the decision space, and $F(\mathbf{x})$ is an objective space.

Definition for Pareto Optimal Set. Consider

$$P = \{x \in \Omega \mid \neg \exists x' \in \Omega \text{ st } F(x') < F(x)\}. \quad (\text{A.1})$$

Definition for Pareto Front. Consider

$$PF = \{F(x) \mid x \in P\}. \quad (\text{A.2})$$

Definition for Nondominated Sorting. Nondominated sorting is used to confirm the rank of each individual in the whole population according to the dominated space. For example, $\forall x_i \in \Omega$, we can compute a dominated set of x_i , $P_{x_i} = \{x \in \Omega \mid \forall x' \in \Omega, \text{ st } : F(x') < F(x_i)\}$. The size of P_{x_i} is the rank of x_i , $\text{Rank}(x_i) = |P_{x_i}|$. We divide the population into different Pareto Front with the rank of each individual, $\Omega = \{F_1 \cup F_2 \cup \dots \cup F_{\text{max_rank}}\}$.

Definition for Crowding Distance. The crowding distance approaches aim to obtain a uniform spread of solutions along a similar Pareto Front without using a fitness sharing parameter. In this paper, there are two objectives, that is, revenue and robustness. Firstly, we rank all individuals in order according to the revenue and then compute the distance of each individual in the sequence by formula $\text{dis}_k(x_i) =$

$(f(x_{i+1}) - f(x_{i-1})) / (f_k^{\max} - f_k^{\min})$. Thus, the crowding distance can be defined as

$$\begin{aligned} \text{Distance}(x_i) = & \frac{f_{\text{revenue}}(x_{i+1}) - f_{\text{revenue}}(x_{i-1})}{f_{\text{revenue}}^{\max} - f_{\text{revenue}}^{\min}} \\ & + \frac{f_{\text{robust}}(x_{i+1}) - f_{\text{robust}}(x_{i-1})}{f_{\text{robust}}^{\max} - f_{\text{robust}}^{\min}}. \end{aligned} \quad (\text{A.3})$$

Definition for Fitness Function. The determination of an appropriate fitness function is crucial to the performance of the genetic algorithm. Thus, we should construct optimization objectives with a weight sum to represent the fitness function [46]:

$$\begin{aligned} F = & w_{\text{revenue}} \sum_{i=1}^N \sum_{j=1}^M \sum_{k=1}^{K_{ij}} x_{i,k}^j p_i \\ & + w_{\text{robustness}} \sum_{i=1}^N \sum_{j=1}^M \sum_{k=1}^{K_{ij}} p_i \times x_{i,k}^j \times \text{Rearrange}(t_i), \end{aligned} \quad (\text{A.4})$$

where F is the fitness and w_{revenue} and $w_{\text{robustness}}$ are weights.

B. The Compact Task Merging Strategy

The compact task merging strategy, which is embedded in the rule-based heuristic algorithm, is used to improve the imaging efficiency. If two or more targets are geographically adjacent, we can rationally tune the slewing angle and the

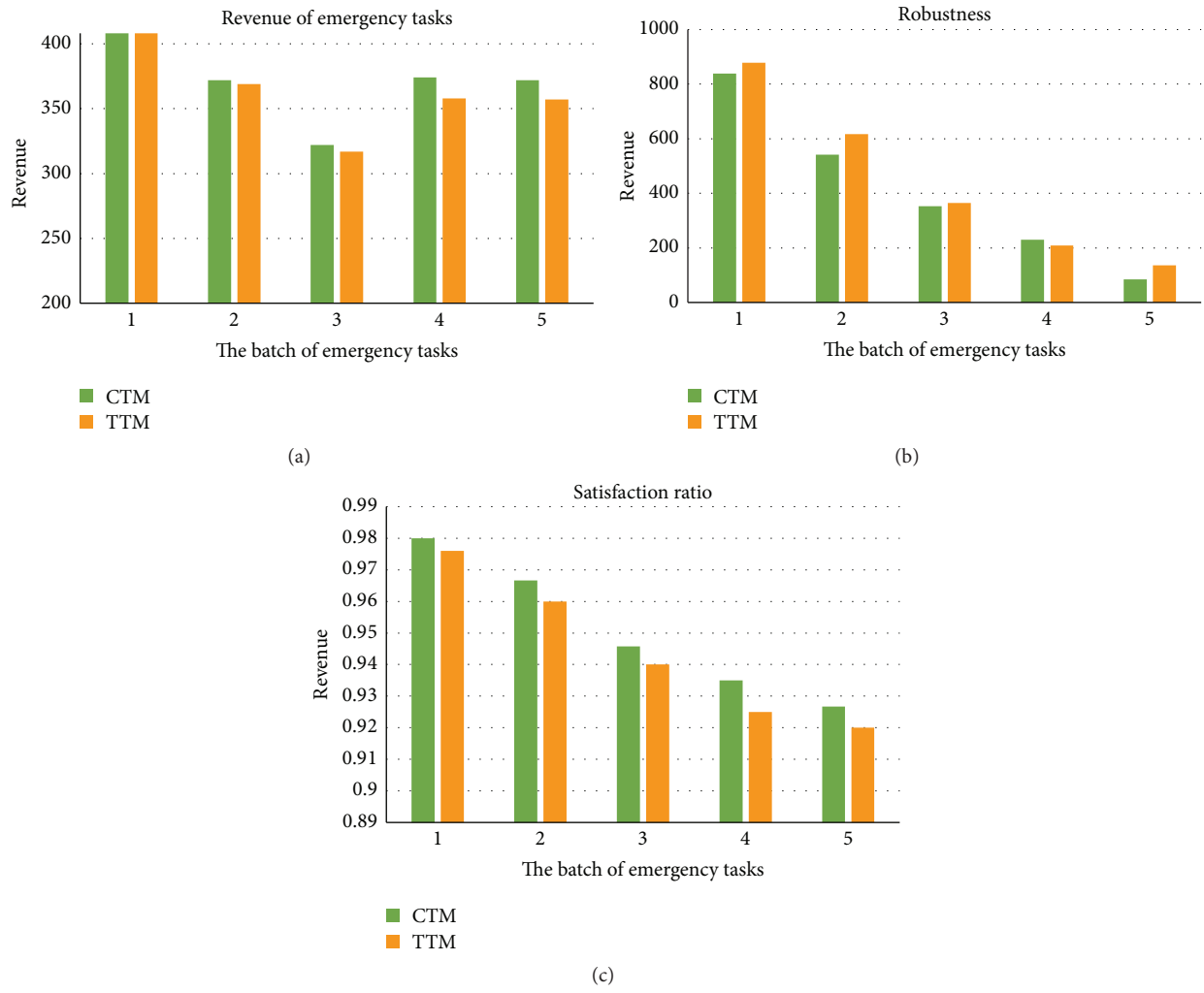


FIGURE 16: The performances of different task merging methods.

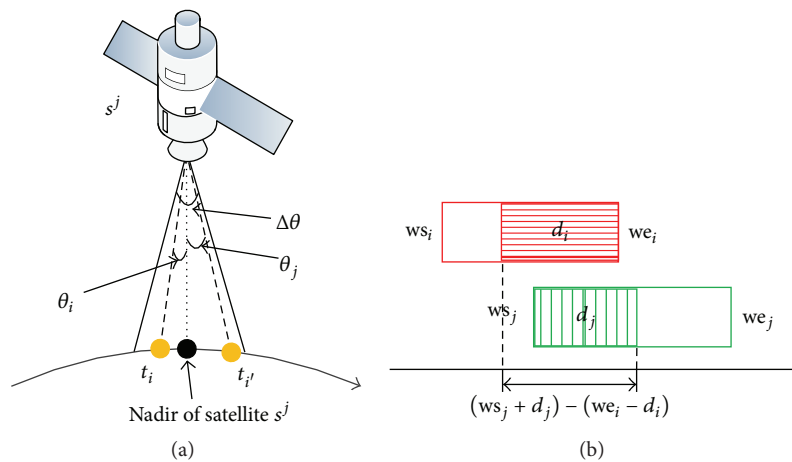


FIGURE 17: Task merging constraints.

observation duration of the sensor to enable an observation strip to cover them. In other words, tasks in the same swath of a sensor may be merged into one composite task. To improve the imaging opportunities of the new tasks, a task merging strategy is required. Suppose there are two tasks t_i and t_j that could be imaged by satellite j . And $ao_i = \{[ws_i, we_i], \theta_i\} \in AO_i^j$ is an available opportunity of task t_i . Accordingly, $ao_j = \{[ws_j, we_j], \theta_j\} \in AO_j^j$ is an available opportunity of task t_j . Since the length of a visible time window must be larger than the observation duration of task, there often exists some unnecessary time to finish merged tasks according to the traditional task merging strategy. Therefore, the duration of task execution is an important factor in task merging. By considering the duration of task execution, we employ a new compact task merging method to construct the compact composite tasks in this paper.

When a task merging mechanism is embedded into the schedule scheme, we must judge when two tasks can be combined into a composite task and determine how to construct a composite task.

Without loss of generality, between two tasks t_i and t_j , the window start time ws_i of task t_i is assumed to be no later than that of task t_j in the following.

Theorem B.1. *Two feasible tasks t_i and t_j can be combined into a compact composite task $t_{i,j}$ if and only if they satisfy*

$$(ws_j + d_j) - (we_i - d_i) \leq \Delta d, \quad (B.1)$$

$$|\theta_i - \theta_j| \leq \Delta \theta. \quad (B.2)$$

Equation (B.1) is the time window constraint. As shown in Figure 17(b), we illustrate the case where two time windows of tasks intersect.

Equation (B.2) is the slewing angle constraint. It is shown in Figure 17(a) that tasks t_i and t_j must be located in the same swath of the sensor.

Theorem B.2. *If two feasible tasks t_i and t_j can be merged into a compact composite task $t_{i,j}$, then its time window $W_{i,j} = [ws_{i,j}, we_{i,j}]$ should range from*

$$ws_{i,j} = \begin{cases} we_i - d_i, & \text{if } |W_i \cap W_j| \leq \min(d_i, d_j) \\ \min\{ws_j, \max(ws_i, ws_j + d_j - d_i)\}, & \text{else} \end{cases} \quad (B.3)$$

to

$$we_{i,j} = \begin{cases} w_j + d_j, & \text{if } |W_i \cap W_j| \leq \min(d_i, d_j) \\ \min(we_i - d_i, we_j - d_j) + \max(d_i, d_j), & \text{else.} \end{cases} \quad (B.4)$$

Its indispensable duration of task execution should be

$$d_{i,j} = \begin{cases} ws_j + d_j - (we_i - d_i), & \text{if } |W_i \cap W_j| \leq \min(d_i, d_j) \\ \max(d_i, d_j), & \text{else} \end{cases} \quad (B.5)$$

and the slewing angle is given by

$$\theta_{i,j} = \begin{cases} \max\left\{\theta_i - \frac{\Delta \theta_s}{2}, 0\right\}, & \text{if } \theta_i \geq 0, |\theta_i| \geq |\theta_j| \\ \min\left\{\theta_i + \frac{\Delta \theta_s}{2}, 0\right\}, & \text{if } \theta_i < 0, |\theta_i| \geq |\theta_j| \\ \max\left\{\theta_j - \frac{\Delta \theta_s}{2}, 0\right\}, & \text{if } \theta_j \geq 0, |\theta_j| > |\theta_i| \\ \min\left\{\theta_j + \frac{\Delta \theta_s}{2}, 0\right\}, & \text{if } \theta_j < 0, |\theta_j| > |\theta_i|. \end{cases} \quad (B.6)$$

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work is supported by the National Basic Research Program of China (no. 2011CB707102), the National Natural Science Foundation of China (no. 41571334), and the Fundamental Research Funds for the Central Universities.

References

- [1] N. Bianchessi, J.-F. Cordeau, J. Desrosiers, G. Laporte, and V. Raymond, "A heuristic for the multi-satellite, multi-orbit and multi-user management of Earth observation satellites," *European Journal of Operational Research*, vol. 177, no. 2, pp. 750–762, 2007.
- [2] Q. Dishan, H. Chuan, L. Jin, and M. Manhao, "A dynamic scheduling method of earth-observing satellites by employing rolling horizon strategy," *The Scientific World Journal*, vol. 2013, Article ID 304047, 11 pages, 2013.

- [3] P. Wang, G. Reinelt, P. Gao, and Y. Tan, "A model, a heuristic and a decision support system to solve the scheduling problem of an earth observing satellite constellation," *Computers & Industrial Engineering*, vol. 61, no. 2, pp. 322–335, 2011.
- [4] G. Wu, M. Ma, J. Zhu, and D. Qiu, "Multi-satellite observation integrated scheduling method oriented to emergency tasks and common tasks," *Journal of Systems Engineering and Electronics*, vol. 23, no. 5, pp. 723–733, 2012.
- [5] N. G. Hall and M. J. Magazine, "Maximizing the value of a space mission," *European Journal of Operational Research*, vol. 78, no. 2, pp. 224–241, 1994.
- [6] E. Bensana, G. Verfaillie, J. Agnese, N. Bataille, and D. Blumstein, "Exact & INEXACT methods for daily management of earth observation satellite," in *Proceedings of the Space Mission Operations and Ground Data Systems (SpaceOps '96)*, pp. 394–507, 1996.
- [7] M. A. A. Mansour and M. M. Dessouky, "A genetic algorithm approach for solving the daily photograph selection problem of the SPOT5 satellite," *Computers & Industrial Engineering*, vol. 58, no. 3, pp. 509–520, 2010.
- [8] D. Habet, M. Vasquez, and Y. Vimont, "Bounding the optimum for the problem of scheduling the photographs of an agile earth observing satellite," *Computational Optimization and Applications*, vol. 47, no. 2, pp. 307–333, 2010.
- [9] L. Barbulescu, A. E. Howe, J. P. Watson, and L. D. Whitley, "Satellite range scheduling: a comparison of genetic, heuristic and local search," in *Parallel Problem Solving from Nature—PPSN VII*, vol. 2439 of *Lecture Notes in Computer Science*, pp. 611–620, Springer, Berlin, Germany, 2002.
- [10] M. Lemaître, G. Verfaillie, F. Jouhaud, J.-M. Lachiver, and N. Bataille, "Selecting and scheduling observations of agile satellites," *Aerospace Science and Technology*, vol. 6, no. 5, pp. 367–381, 2002.
- [11] W.-C. Lin and S.-C. Chang, "Hybrid algorithms for satellite imaging scheduling," in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, vol. 3, pp. 2518–2523, IEEE, Waikoloa Village, Hawaii, USA, October 2005.
- [12] W.-C. Lin, D.-Y. Liao, C.-Y. Liu, and Y.-Y. Lee, "Daily imaging scheduling of an earth observation satellite," *IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans*, vol. 35, no. 2, pp. 213–223, 2005.
- [13] W.-C. Lin and D.-Y. Liao, "A tabu search algorithm for satellite imaging scheduling," in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics (SMC '04)*, pp. 1601–1606, October 2004.
- [14] W. J. Wolfe and S. E. Sorensen, "Three scheduling algorithms applied to the earth observing systems domain," *Management Science*, vol. 46, no. 1, pp. 148–168, 2000.
- [15] V. Gabrel and C. Murat, "Mathematical programming for earth observation satellite mission planning," in *Operations Research in Space and Air*, pp. 103–122, Springer, New York, NY, USA, 2003.
- [16] M. Vasquez and J.-K. Hao, "A 'logic-constrained' knapsack formulation and a tabu algorithm for the daily photograph scheduling of an earth observation satellite," *Computational Optimization and Applications*, vol. 20, no. 2, pp. 137–157, 2001.
- [17] M. Vasquez and J.-K. Hao, "Upper bounds for the SPOT 5 daily photograph scheduling problem," *Journal of Combinatorial Optimization*, vol. 7, no. 1, pp. 87–103, 2003.
- [18] M. Zweben, E. Davis, B. Daun, and M. J. Deale, "Scheduling and rescheduling with iterative repair," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 23, no. 6, pp. 1588–1596, 1993.
- [19] J. Frank, A. Jonsson, R. Morris, and D. E. Smith, "Planning and scheduling for fleets of earth observing satellites," in *Proceedings of the 6th International Symposium on Artificial Intelligence, Robotics, Automation and Space*, Montreal, Canada, June 2001.
- [20] N. Bianchessi and G. Righini, "Planning and scheduling algorithms for the COSMO-SkyMed constellation," *Aerospace Science and Technology*, vol. 12, no. 7, pp. 535–544, 2008.
- [21] P. Wang and G. Reinelt, "A heuristic for an earth observing satellite constellation scheduling problem with download considerations," *Electronic Notes in Discrete Mathematics*, vol. 36, pp. 711–718, 2010.
- [22] F. Marinelli, S. Nocella, F. Rossi, and S. Smriglio, "A Lagrangian heuristic for satellite range scheduling with resource constraints," *Computers & Operations Research*, vol. 38, no. 11, pp. 1572–1583, 2011.
- [23] L. Barbulescu, J.-P. Watson, L. D. Whitley, and A. E. Howe, "Scheduling space-ground communications for the air force satellite control network," *Journal of Scheduling*, vol. 7, no. 1, pp. 7–34, 2004.
- [24] J. Wang, N. Jing, J. Li, and Z. H. Chen, "A multi-objective imaging scheduling approach for earth observing satellites," in *Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation (GECCO '07)*, pp. 2211–2218, London, UK, 2007.
- [25] S.-W. Baek, S.-M. Han, K.-R. Cho et al., "Development of a scheduling algorithm and GUI for autonomous satellite missions," *Acta Astronautica*, vol. 68, no. 7-8, pp. 1396–1402, 2011.
- [26] Y. Chen, D. Zhang, M. Zhou, and H. Zou, "Multi-satellite observation scheduling algorithm based on hybrid genetic particle swarm optimization," in *Advances in Information Technology and Industry Applications*, vol. 136 of *Lecture Notes in Electrical Engineering*, pp. 441–448, Springer, Berlin, Germany, 2012.
- [27] Z. Zhang, N. Zhang, and Z. Feng, "Multi-satellite control resource scheduling based on ant colony optimization," *Expert Systems with Applications*, vol. 41, no. 6, pp. 2816–2823, 2014.
- [28] X. Liu, B. Bai, Y. Chen, and Y. Feng, "Multi satellites scheduling algorithm based on task merging mechanism," *Applied Mathematics and Computation*, vol. 230, pp. 687–700, 2014.
- [29] G. Wu, J. Liu, M. Ma, and D. Qiu, "A two-phase scheduling method with the consideration of task clustering for earth observing satellites," *Computers & Operations Research*, vol. 40, no. 7, pp. 1884–1894, 2013.
- [30] D. Qiu, L. Zhang, J. Zhu, and H. Li, "FFFS-DTMB and ADTPC-DTMB algorithm in multi-satellites mission planning," *Acta Aeronautica et Astronautica Sinica*, vol. 30, pp. 2178–2184, 2009.
- [31] A. Globus, J. Crawford, J. Lohn, and A. Pryor, "A comparison of techniques for scheduling earth observing satellites," in *Proceedings of the 16th Conference on Innovative Applications of Artificial Intelligence (IAAI '04)*, pp. 836–843, AAAI, San Jose, Calif, USA, July 2004.
- [32] A. Globus, J. Crawford, J. Lohn, and A. Pryor, "Scheduling earth observing satellites with evolutionary algorithms," in *Proceedings of the International Conference on Space Mission Challenges for Information Technology (SMC-IT '03)*, Pasadena, Calif, USA, July 2003.
- [33] J. C. Pemberton and L. G. Greenwald, "On the need for dynamic scheduling of imaging satellites," *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 34, pp. 165–171, 2002.
- [34] L. A. Kramer and S. F. Smith, "Maximizing flexibility: a retraction heuristic for oversubscribed scheduling problems,"

- in *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI '03)*, pp. 1218–1223, August 2003.
- [35] G. Verfaillie and T. Schiex, “Solution reuse in dynamic constraint satisfaction problems,” in *Proceedings of the 12th National Conference on Artificial Intelligence (AAAI '94)*, pp. 307–312, Seattle, Wash, USA, July 1994.
 - [36] J. Wang, J. Li, and Y. Tan, “Study on heuristic algorithm for dynamic scheduling problem of earth observing satellites,” in *Proceedings of the 8th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD '07)*, pp. 9–14, Qingdao, China, July 2007.
 - [37] J. Wang, X. Zhu, L. T. Yang, J. Zhu, and M. Ma, “Towards dynamic real-time scheduling for multiple earth observation satellites,” *Journal of Computer and System Sciences*, vol. 81, no. 1, pp. 110–124, 2015.
 - [38] M. Wang, G. Dai, and M. Vasile, “Heuristic scheduling algorithm oriented dynamic tasks for imaging satellites,” *Mathematical Problems in Engineering*, vol. 2014, Article ID 234928, 11 pages, 2014.
 - [39] J. Wang, X. Zhu, D. Qiu, and L. T. Yang, “Dynamic scheduling for emergency tasks on distributed imaging satellites with task merging,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 9, pp. 2275–2285, 2014.
 - [40] B.-C. Bai, Y.-Z. Ci, and Y.-W. Chen, “Dynamic task merging in multi-satellites observing scheduling,” *Journal of System Simulation*, vol. 21, no. 9, pp. 2646–2649, 2009.
 - [41] J. Branke, “Creating robust solutions by means of evolutionary algorithms,” in *Parallel Problem Solving from Nature—PPSN V*, vol. 1498 of *Lecture Notes in Computer Science*, pp. 119–128, Springer, Berlin, Germany, 1998.
 - [42] S. Tsutsui and A. Ghosh, “Genetic algorithms with a robust solution searching scheme,” *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 3, pp. 201–208, 1997.
 - [43] F. Khafa, J. Sun, A. Barolli, A. Biberaj, and L. Barolli, “Genetic algorithms for satellite scheduling problems,” *Mobile Information Systems*, vol. 8, no. 4, pp. 351–377, 2012.
 - [44] X. Niu, H. Tang, L. Wu, R. Deng, and X. Zhai, “Imaging-duration embedded dynamic scheduling of Earth observation satellites for emergent events,” *Mathematical Problems in Engineering*, vol. 2015, Article ID 731734, 31 pages, 2015.
 - [45] T. Mao, Z. Xu, R. Hou, and M. Peng, “Efficient satellite scheduling based on improved vector evaluated genetic algorithm,” *Journal of Networks*, vol. 7, no. 3, pp. 517–523, 2012.
 - [46] B. Sun, W. Wang, X. Xie, and Q. Qin, “Satellite mission scheduling based on genetic algorithm,” *Kybernetes*, vol. 39, 8, pp. 1255–1261, 2010.
 - [47] M. T. Jensen, “Generating robust and flexible job shop schedules using genetic algorithms,” *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 3, pp. 275–288, 2003.

