Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 731734, 31 pages
http://dx.doi.org/10.1155/2015/731734

Research Article

Imaging-Duration Embedded Dynamic Scheduling of Earth Observation Satellites for Emergent Events

Xiaonan Niu,[1,2] Hong Tang,[1,2] Lixin Wu,[3] Run Deng,[1,2] and Xuejun Zhai[1,2]

[1]*State Key Laboratory of Earth Surface Processes and Resource Ecology, Beijing Normal University, Beijing 100875, China*
[2]*Key Laboratory of Environment Change and Natural Disaster, Ministry of Education, Beijing Normal University, Beijing 100875, China*
[3]*School of Environment Science and Spatial Informatics, China University of Mining and Technology, Xuzhou 221116, China*

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

Received 22 December 2014; Revised 9 May 2015; Accepted 13 May 2015

Academic Editor: Ivanka Stamova

We present novel two-stage dynamic scheduling of earth observation satellites to provide emergency response by making full use of the duration of the imaging task execution. In the first stage, the multiobjective genetic algorithm NSGA-II is used to produce an optimal satellite imaging schedule schema, which is robust to dynamic adjustment as possible emergent events occur in the future. In the second stage, when certain emergent events do occur, a dynamic adjusting heuristic algorithm (CTM-DAHA) is applied to arrange new tasks into the robust imaging schedule. Different from the existing dynamic scheduling methods, the imaging duration is embedded in the two stages to make full use of current satellite resources. In the stage of robust satellite scheduling, total task execution time is used as a robust indicator to obtain a satellite schedule with less imaging time. In other words, more imaging time is preserved for future emergent events. In the stage of dynamic adjustment, a compact task merging strategy is applied to combine both of existing tasks and emergency tasks into a composite task with least imaging time. Simulated experiments indicate that the proposed method can produce a more robust and effective satellite imaging schedule.

1. Introduction

Recently, earth observing satellites (EOSs) are widely used in applications for national defense, environmental protection, agriculture, meteorology, urban construction, and other fields. However, satellite resources are still scarce with respect to the increasing human demands for imaging. As a result, the process of satellite mission scheduling, which is used to allocate the observation resources and execution time to a series of imaging tasks by maximizing one or more objectives while satisfying certain given constraints, plays an important role in the management of satellites. The scheduling can be primarily divided into static scheduling and dynamic scheduling. The 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 of several unexpected factors, such as a thick cloud cover, resource changes, and new tasks arrival, the initial scheduling scheme must be adjusted dynamically; such scheduling is called dynamic scheduling. However, either static scheduling or dynamic scheduling is a complex combination optimization problem that has been proved to be NP-complete [1].

Over the last several decades, development of methods to perform satellite mission scheduling has been intensively investigated, most of which are focused on the static scheme. The algorithms to solve the problem can be mainly divided into exact methods and approximate methods. The approximate methods include the intelligent optimization algorithms and rule-based heuristic algorithms. The exact methods, such as dynamic programming, the branch-and-bound algorithm, and the Russian Doll Search, were used mostly at the early stage of satellite scheduling. Verfaillie et al. viewed earth observation satellite scheduling as a valued constraint satisfaction problem and developed the Russian Doll Search to solve it [2]. Ovacik and Uzsoy decomposed

the scheduling problem into many subproblems and solved these subproblems to the optimality by a branch-and-bound algorithm [3]. Bensana et al. applied a number of global search approaches, including the depth first branch-and-bound algorithm, the best first branch-and-bound algorithm, and the Russian Doll Search, to obtain the solution for the Spot5 daily scheduling problem [4]. The exact methods can provide optimal solutions. However, these exact methods can only solve small-scaled problems. The approximate methods, that is, the intelligent optimization algorithms and rule-based heuristic algorithms, are aimed at identifying good solutions that may not be optimal. The intelligent optimization algorithms primarily included the Tabu search algorithm, the genetic algorithm, the evolutionary algorithm, simulated annealing, the Lagrangian relaxation technique, and the hybrid ant colony optimization method. Vasquez and Hao translated the scheduling problem into the well-known knapsack model. They proposed a Tabu search algorithm to solve the model [5]. Bianchessi et al. investigated the scheduling problem for a constellation of agile satellites. A Tabu search algorithm was devised to produce solutions [6]. Baek et al. applied a new genetic algorithm for simulations of an actual satellite mission scheduling problem [7]. Mansour and Dessouky developed a genetic algorithm for solving the scheduling problem using a new genome representation for maximizing multicriteria objectives including the profit and the number of acquired photographs [8]. Globus et al. hypothesized that evolutionary algorithms can effectively schedule coordinated fleets of earth observing satellites and compared the evolutionary algorithm and other methods to test the hypothesis [9]. Wang et al. proposed a multiobjective EOS imaging scheduling method based on the Strength Pareto Evolutionary Algorithm-II [10]. Lin et al. adopted the Lagrangian relaxation and linear search techniques to solve the daily imaging scheduling problem to acquire a near-optimal solution [11]. Wu et al. proposed a hybrid ant colony optimization mixed with local search to obtain satisfactory schedules to address the satellite observation scheduling problem [12]. Zhang et al. presented an algorithm for a multisatellite control resource scheduling problem based on ant colony optimization [13]. These intelligent algorithms, as mentioned above, can be used to obtain near-optimal solutions for large size problems. In addition, rule-based heuristic algorithms have been used to solve the satellite scheduling. Hall and Magazine designed eight heuristic methods for selecting and scheduling projects to maximize the value of a space mission. The computational tests revealed that these methods routinely delivered very close to optimal solutions [1]. Wang et al. presented a nonlinear model of the scheduling problem and developed a priority-based heuristic with conflict-avoided, limited backtracking and download-as-needed features to solve it. They found the heuristic method can produce satisfactory and feasible plans in a notably short time [14]. The rule-based heuristic methods are more flexible approaches to obtain satisfactory solutions that are close to optimal solutions. To summarize, the approximate methods can provide near-optimal solutions to large-scaled problems.

However, all of the above research studies only focused on common tasks in a static environment. Once a schedule is made, it cannot be changed, which is not feasible in dynamic environment. For example, when an earthquake occurs, new emergency tasks with high priority are very difficult to insert into the scheduling scheme. Therefore, determining how to schedule new tasks dynamically is critical. The general method of recent research is to produce a temporary schedule and then to adjust the schedule as quickly as possible while maintaining the efficiency and stability of the schedule as well. Verfaillie and Schiex modeled the dynamic satellite scheduling as a dynamic constraint satisfaction problem. They proposed a new method by reusing any previous solution and producing a new one by local changes on the previous one [15]. Wu et al. used a hybrid ant colony optimization method mixed with iteration local search to obtain a schedule. Next, they proposed a repair method to schedule emergency tasks [16]. Qiu et al. decomposed scheduling horizon into a series of static scheduling intervals and used a rolling horizon strategy to optimize the scheduling schemes in each interval [17]. Sun et al. described the dynamic scheduling problem as a dynamic weighted maximal constraint satisfaction problem in which constraints can be changed dynamically [18]. Wang et al. analyzed the dynamic properties of satellite scheduling and proposed two heuristic algorithms to schedule new tasks [19]. Wang et al. described the dynamic scheduling problem with a unified form of inserting new tasks. Concentrating on how to insert new tasks in initial schedule, they proposed a rule-based heuristic algorithm [20]. Wang et al. focused on how to insert new tasks dynamically in a schedule. These researchers presented a new dynamic real-time scheduling algorithm considered a task dynamic merging strategy [21].

Unfortunately, to the best of our knowledge, no work has been done with respect to the duration of task execution. In addition, less work considered a task merging mechanism in dynamic scheduling. Although some traditional merging methods were studied in a few of previous researches [12, 22–25], these methods did not take into account the duration of task execution. As is known, the duration of task execution indicates how long an imaging task must be observed practically. Since the length of the available visible time windows must be larger than the duration of the task, there often exists some unnecessary time to finish the task. With the consideration of the duration of task execution, more spare time in the schedule will exist, which may provide more imaging opportunities for new tasks. In addition, the task merging strategy using the duration of task can improve the number of tasks for the satellite to finish, thereby enabling many more new tasks to be assigned to an initial schedule.

In this paper, we present a novel two-stage method for dynamic scheduling of earth observation satellites to address emergent events by making full use of the duration of imaging task execution. The method is comprised of two stages: robust satellite scheduling and dynamic adjustment. In the first stage, we establish a robust satellite scheduling model that accounts for the total task execution time and use the multiobjective genetic algorithm NSGA-II to create feasible initial schemes. In the second stage, we adjust the robust

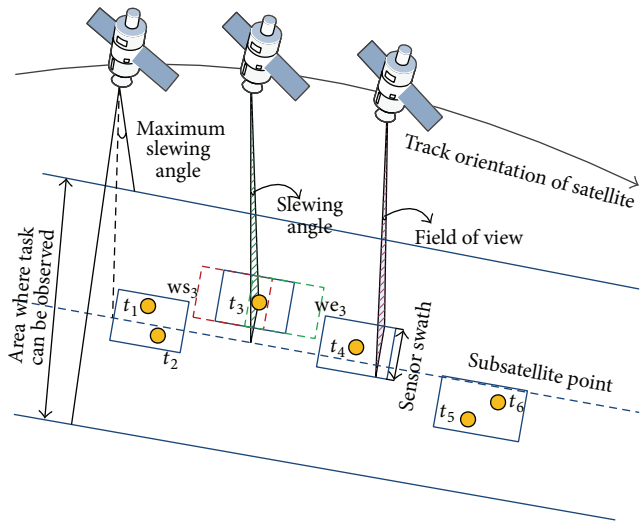

Figure 1: Illustration of satellite observing activity.

solution to insert new tasks. The dynamic adjusting rule-based heuristic algorithm (CTM-DAHA) is designed to get a satisfactory schedule which generates high revenue and little disturbance. To improve the imaging efficiency as much as possible, we propose to embed a compact composite task merging method that considers task execution time into the algorithm.

The major contributions of this paper are summarized as follows. (1) For the first time, the total task execution time is regarded as an indicator to evaluate the robustness of the scheduling schemes. (2) A compact task merging method that considers the duration of task execution is embedded into the dynamic scheduling algorithm.

The remainder of the paper is organized as follows. The dynamic scheduling problem as well as the two-stage solution framework is described in Section 2. In Section 3, we present the robust satellite scheduling model and algorithm. In Section 4, we propose a new heuristic algorithm considering a compact task merging mechanism to dynamically adjust the initial schedule. In Section 5, we conduct experimental simulations and compare different algorithms used for scheduling. We conclude the paper with a summary in Section 6.

2. Problem Formulation

In current section, we will firstly introduce the process of satellite observation and task merging method in brief. Then we describe the problem of dynamic scheduling oriented emergent events. Moreover, the framework of two-stage dynamic scheduling method is presented.

2.1. Description of Satellite Observation. 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, when subject to related constraints. As shown in Figure 1, the EOS operates in space in a certain orbit. When the EOS flies over the target area, its sensor is opened to take the image. We assume that the sensors of the EOSs considered in our study are able to slew laterally. A target is termed as a task in this paper. As the imaging process will last a few seconds, it will produce a strip that covers the target. The strip of EOS can be formed on the ground by the subsatellite point of satellite as well as the field of view of the sensor, the slewing angle of the sensor, and the observation duration. The observation duration indicates how long an imaging task must be observed practically. It depends on the satellite's travelling speed, the sensor's scanning speed, and the ground strip to be scanned. A task must be imaged by the satellite within the available time windows. Taking as an example task t_3, at the moment $t = \text{ws}$, task t_3 begins to appear in the scope of the EOS, and, with the movement of the EOS, t_3 disappears at the moment $t = \text{we}$. Therefore, the EOS can observe task t_3 between ws and we; that is, $[\text{ws}, \text{we}]$ is a time window of t_3. The time windows (as well as slewing angle) between the satellite and the task can be computed based on orbit parameters. Because the length of the available visible time window must be larger than the duration of the task, there exists unnecessary time within the window to finish the task. With the consideration of the duration of task execution, more spare time in the schedule will exist, which may provide more imaging opportunities for new tasks. Therefore, we view the duration of task execution as an important factor.

If two or more targets are geographically adjacent, we can rationally tune the slewing angle and the observation duration of the sensor to enable an observation strip to cover them. In other words, multiple tasks can be merged into a composite task. Tasks t_1 and t_2 can be merged and completed by one observation activity just as Figure 1 shows. Although there are some traditional merging methods in a few previous researches which considered a task merging

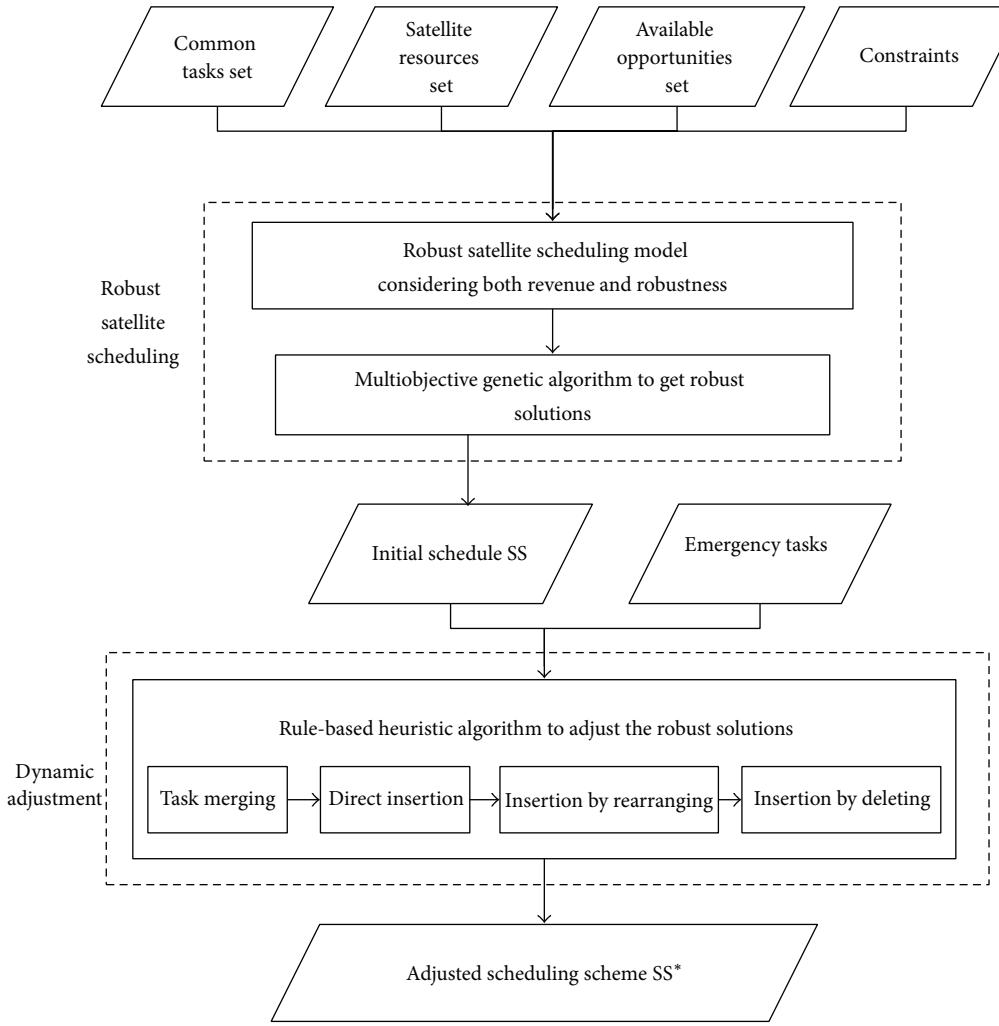

Figure 2: The framework of the two-stage dynamic scheduling approach.

mechanism [12, 22–25], the duration of task execution, to the best of our knowledge, is out of consideration. By considering the duration of task execution, we propose a new compact task merging method to construct the so-called compact composite tasks in this paper. Specifically, a compact composite task is characterized by the smallest slewing angle, the shortest duration of task execution, and the most compact time window. We describe the details of compact task merging method in Section 4.2.

2.2. Dynamic Scheduling. The static scheduling problem is focused on common tasks, which primarily are periodic in a static environment. Once a schedule is made, it cannot be changed until all of the scheduled tasks are completed. Compared to the problem of static satellite mission scheduling, dynamic scheduling oriented emergent events means that there are a number of emergency tasks that must be allocated dynamically. Dynamic scheduling is a more complex combination optimization problem. A general solution to the problem is dynamically adjusting the initial scheduling scheme SS which is generated in advance to produce a new schedule SS* containing new tasks. In addition to the objective of high revenue, we have to consider the robustness of the scheduling scheme SS. Usually the dynamic scheduling problem needs to satisfy two objectives: to maximize the revenue and to minimize the perturbation to the previous scheduling [19, 20].

To maintain high revenue of the adjusted scheduling scheme and to minimize the difference between the adjusted schedule and the initial schedule, we account for the robustness of the schedule and propose a two-stage method, which includes robust satellite scheduling and dynamic adjustment. In each stage, the duration of task execution is used.

As shown in Figure 2, the proposed approach to the dynamic scheduling of satellites consists of two stages: robust satellite scheduling and dynamic adjustment. In the robust satellite scheduling stage, we establish a robust satellite scheduling model with three objectives considering both revenue and robustness. Two of the objectives are related to robustness: optimization of the total task execution time and

optimization of the neighborhood-based robust indicator. To solve the model, the multiobjective genetic algorithm NSGA-II is used to obtain robust solutions. In the dynamic adjustment stage, with the arrival of emergency tasks, we dynamically adjust the initial scheduling scheme SS generated in the first stage. Four methods are used to insert new tasks: compact task merging considering the duration of task execution, direct insertion, insertion by rearranging, and insertion by deleting. Finally, a rule-based heuristic algorithm is designed to get the adjusted scheduling scheme SS^*.

3. Robust Satellite Scheduling

Focusing on the common tasks, the robust satellite scheduling model considering both revenue and robustness is constructed in the first stage. It is worth mentioning that we view the total task execution time as another objective to evaluate the robustness of schedule. The multiobjective genetic algorithm NSGA-II is applied to solve the model to get feasible initial schedules.

3.1. Model. Given a set of imaging tasks on multiple satellites, a satellite scheduling scheme consists of a subset of tasks to be executed on specific satellites with specific time spans by maximizing objectives while simultaneously satisfying some given constraint conditions. As a result, the satellite scheduling problem usually consists of five parts: tasks, satellite resources, opportunities, objectives, and constraints. In the first stage, the robust satellite scheduling model built in the paper consists of three objectives, that is, maximum revenue $R(\mathrm{SS})$, maximum value of the neighborhood-based robust indicator $N(\mathrm{SS})$, and minimum total duration of task execution $T(\mathrm{SS})$, and four constraints, that is, uniqueness constraint, switch time constraint, time window constraint, and imaging time constraint.

(1) Tasks. Consider $T = \{t_1, t_2, \ldots, t_N\}$, where N is the number of tasks. Each task $t_i \in T$ has a weight p_i, an indispensable duration of task execution d_i. We assume that all tasks are point targets; that is, each target is viewed as a point that can be observed by a single observations trip.

(2) Satellite Resources. Consider $S = \{s^1, s^2, \ldots, s^M\}$, where M is the satellite number. Each satellite $s^j \in S$ can be denoted by $s^j = (\Delta\theta^j, \Delta d^j, \mathrm{sl}^j, \mathrm{st}^j, \mathrm{msg}^j, \mathrm{orb}^j, \mathrm{duty}^j)$ to describe its observation capability. The related notations are defined as follows:

$\Delta\theta^j$: the field of view (FOV), which describes the angle range that a lens of the sensor can image;

Δd^j: the longest duration allowed for a continuous observation;

sl^j: slewing rate, that is, the time consumed for each angle of satellite slewing;

st^j: attitude stability time, which describes the required time after adjusting the posture of satellite j;

msg^j: maximum slewing angle, which reflects the observation capability of the satellite j;

orb^j: the duration satellite j that circles the earth each time;

duty^j: the longest time for satellite j opening its sensor in each orbit.

(3) Available Opportunities. The opportunities include time windows and slewing angles. $\mathrm{AO}_i^j = \{\mathrm{ao}_{i1}^j, \mathrm{ao}_{i2}^j, \ldots, \mathrm{ao}_{iK_{ij}}^j\}$ is the set of available opportunities of task t_i on satellite s^j. For a given available opportunity ao_{ik}^j, it is represented by $\mathrm{ao}_{ik}^j = \{[\mathrm{ws}_{ik}^j, \mathrm{we}_{ik}^j], \theta_{ik}^j\}$, where ws_{ik}^j and we_{ik}^j denote the start time and end time, respectively, of the time window W_{ik}^j and θ_{ik}^j is the ideal slewing angle, as depicted in Figure 1. $W_i^j = \cup_{k\in[1,\ldots,K_{ij}]} W_{ik}^j$ is the set of time windows of task i on satellite j.

(4) Objectives. There are three objectives, that is, maximum revenue $R(\mathrm{SS})$, maximum value of the neighborhood-based robust indicator $N(\mathrm{SS})$, and minimum total duration of task execution $T(\mathrm{SS})$, in the robust scheduling model built in the paper.

The primary objective is to maximize the revenue measured by the sum of weights of all the scheduled tasks

$$\max : R(\mathrm{SS}) = \sum_{i=1}^{N}\sum_{j=1}^{M}\sum_{k=1}^{K_{ij}} x_{i,k}^j p_i, \tag{1}$$

where $x_{i,k}^j$ is the decision variable that can be either 1 if the task i is executed by the satellite j in the kth time window or 0 otherwise.

Let y_i be the parameter that indicates whether the task $t_i \in \mathrm{SS}$ can be rearranged in another timeslot

$$y_i = \begin{cases} 1, & \text{if task } t_i \text{ can be rearranged in another timeslot} \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

The second objective is called the neighborhood-based robust indicator [26]. We define this objective as the total revenue of the scheduled tasks that can be reassigned into other timeslots in the schedule. This objective measures the ability of a scheduling scheme to rearrange the scheduled tasks. The higher the value of the neighborhood-based robust indicator, the more the possibility for the rearrangement of the scheduled tasks. Therefore, we maximize the value of the neighborhood-based robust indicator

$$\max : N(\mathrm{SS}) = \sum_{i=1}^{N}\sum_{j=1}^{M}\sum_{k=1}^{K_{ij}} p_i \times x_{i,k}^j \times y_i. \tag{3}$$

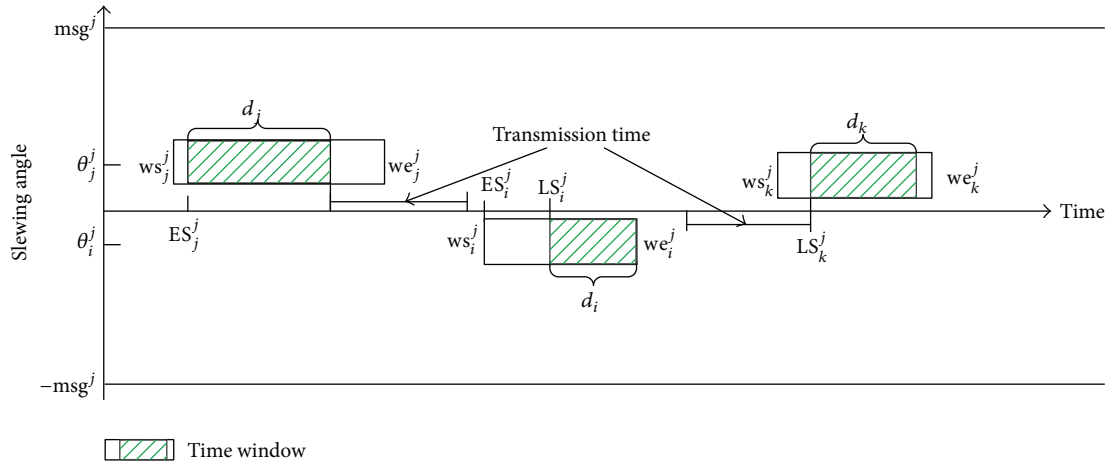

Figure 3: Illustration of the earliest time and the latest time of scheduled task i for satellite j.

Moreover, another robust indicator is the total task execution time. This indicator can reflect the ability of a scheduling scheme to accept new tasks. If the total task execution time is shorter, then there will be more spare time that may provide more imaging opportunities for new tasks. Thus, the last objective is to minimize the total duration of the scheduled tasks

$$\min : T\,(\mathrm{SS}) = \sum_{i=1}^{N}\sum_{j=1}^{M}\sum_{k=1}^{K_{ij}} x_{i,k}^{j} d_i. \tag{4}$$

(5) Constraints. In addition to achieving the objectives, four constraints must be simultaneously satisfied in the model.

Because each task should be performed no more than once by all satellites, therefore, we have the following uniqueness constraint:

$$\sum_{j=1}^{M}\sum_{k=1}^{K_{ij}} x_{i,k}^{j} \le 1. \tag{5}$$

Any two consecutive tasks $t_u, t_v \in T$ assigned to the same satellite s^j should have an adequate transmission time for sensor adjusting and stabilizing gesture. Therefore, we have the switch time constraint

$$\mathrm{ts}_u^j + d_u + s_{u,v}^j \le \mathrm{ts}_v^j, \tag{6}$$

where ts_u^j denotes the beginning time of task t_u and $\mathrm{tr}_{u,v}^j = \mathrm{sl}^j \times |\theta_v^j - \theta_u^j| + \mathrm{st}^j$ denotes the transmission time between tasks t_u and t_v.

The decision variable $x_{i,k}^j$ indicates whether the task i is executed. For each scheduled task in SS, its beginning time and slewing angle must be determined. All the scheduled tasks form a scheduling scheme $\mathrm{SS} = \cup_{j\in[1,M]} T^j$, where T^j is a sequence of the scheduled tasks ordered in time for satellite j. Because the length of the time window of a task must be larger than the duration of the task, the start time of the task is flexible. The beginning time usually ranges from the earliest start time to the latest start time. Taking as an example task $t_i \in T^j$, the start time ts_i^j is related to the time window, the prior task t_j, and the next task t_k of t_i in the sequence T^j, as shown in Figure 3. We assume that the available opportunity $\mathrm{ao}_i^j = \{[\mathrm{ws}_i^j, \mathrm{we}_i^j], \theta_i^j\}$ is chosen to arrange task t_i. Thus, the slewing angle of the task is θ_i^j and the start time of the task is $\mathrm{ts}_i^j \in [\mathrm{ES}_i^j, \mathrm{LS}_i^j]$, where $\mathrm{ES}_i^j = \max(\mathrm{ES}_j^j + d_j + \mathrm{tr}_{j,i}, \mathrm{ws}_i^j)$ is the earliest start time and $\mathrm{LS}_i^j = \min(\mathrm{LS}_k^j - \mathrm{tr}_{i,k} - d_i, \mathrm{we}_i^j - d_i)$ is the latest start time.

If any task is executed, then the execution time should be within its time windows. Hence, we have time window constraint

$$\begin{aligned} x_{i,k}^j \left(\mathrm{ts}_i^j - \mathrm{ws}_{ik}^j\right) &\ge 0, \\ x_{i,k}^j \left(\mathrm{ts}_i^j + d_i - \mathrm{we}_{ik}^j\right) &\le 0. \end{aligned} \tag{7}$$

The total imaging time of any satellite should be less than the allowable longest imaging time of its sensor during any period time of orb^j. Let the scheduling period be $[b, f]$. Hereby, we have the imaging time constraint

$$\sum_{i\in T_{t_b}^j} d_i \le \mathrm{duty}^j, \tag{8}$$

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

3.2. Multiobjective Genetic Algorithm. To address the multiobjective optimization model, the multiobjective genetic algorithm NSGA-II is used in this paper. NSGA-II is a multiobjective genetic algorithm that is based on the Pareto front. The algorithm has a clear direction for the selection of the elite, which must be retained in each generation so that the groups converge rapidly and are distributed evenly.

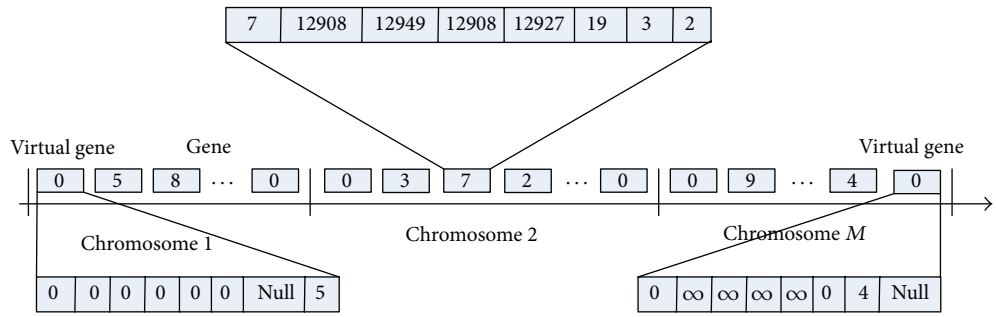

Figure 4: Coding of chromosome.

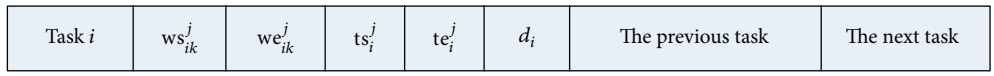

Figure 5: The structure of genes.

Firstly, we make clear some terms that will be referred to in the algorithm. Considering the multiobjective minimization problem in the general form, we define the following definitions [27].

Definition 1 (Pareto dominance). Vector $u = (u_1, u_2, \ldots, u_k)$ superior to vector $v = (v_1, v_2, \ldots, v_k)$ means that $\forall i \in (1, 2, \ldots, k)$ has $u_i < v_i$, and $\exists i \in (1, 2, \ldots, k)$ makes $u_i < v_i$ record as $u_i \prec v_i$, also called vector u dominating vector v.

Definition 2 (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 $F(x') \prec F(x)$. $\mathbf{x}$ is a k-dimensional decision variable in the decision space, and $F(x)$ is an objective space.

Definition 3 (Pareto optimal set). Consider

$$P = \left\{x \in \Omega \mid \neg\exists x' \in \Omega \text{ st } F\left(x'\right) \prec F\left(x\right)\right\}. \tag{9}$$

Definition 4 (Pareto front). Consider

$$\mathrm{PF} = \left\{F\left(x\right) \mid x \in P\right\}. \tag{10}$$

Definition 5 (nondominated sorting). Population Q is divided into different Pareto fronts depending on the dominance degree. Individuals in each subpopulation Q_j are noninferior solutions of this subpopulation. No individual is more dominant than the other individuals. The set of the first front is that Q_1 is the best nondominated set in population Q. The secondary nondominated set in population Q belongs to Q_2, and so on.

Definition 6 (crowding distance). The crowding distance approaches aim to obtain a uniform spread of solutions along the best-known Pareto front without using a fitness sharing parameter. We compute the crowding distance as follows [28].

Step 1. Rank the population and identify the nondominated fronts $F_1, F_2, \ldots, F_R$. For each front $j = 1, 2, \ldots, R$, repeat Steps 2 and 3.

Step 2. For each objective function k, sort the solutions in F_j in the ascending order. Let $l = |F_j|$ and $x_{i,k}$ represent the ith solution in the sorted list with respect to the objective function k. Assign $d_k(x_{1,k}) = \infty$ and $d_k(x_{l,k}) = \infty$, and for $i = 2, 3, \ldots, l-1$ assign

$$d_k\left(x_{i,k}\right) = \frac{f_k\left(x_{i+1,k}\right) - f_k\left(x_{i-1,k}\right)}{f_k^{\max} - f_k^{\min}}. \tag{11}$$

Step 3. To find the total crowding distance $d(x)$ of a solution $\mathbf{x}$, sum the solution's crowding distances with respect to each objective, $d(x) = \sum_k d_k(x)$.

After clarifying these definitions, we describe the components of our algorithm: the coding, the initialization population algorithm, and the genetic operators, that is, the selection operator, the crossover operator, and the mutation operator in the following.

(1) The Coding. The proposed chromosome representation is comprised of a number of genes which is a permutation list of the candidate tasks. The first gene and the last gene on a chromosome are virtual tasks, which are used to mark the position where the chromosome begins and ends. Each gene holds a constant structure to store the information of the corresponding task (Figure 5). Take, for example, the task 7 on chromosome 2. As shown in Figure 4, the allele values of 7, 12908, 12949, 12908, 12927, 19, 3, and 2 represent the task ID, the start time of the time window of the task, the end time of the time window of the task, the start observing time of the task,

the finish observing time of the task, the duration of the task, the previous task, and the next task, respectively.

(2) Initialization Population Algorithm. We use a random generation strategy to generate the initial population.

Step 1. Initialize the task ID $i = 1$ and individual ID $j = 1$. Set M as the size of population and n_t as the number of tasks.

Step 2. Initialize the first individual $I_j = \emptyset$ and then randomly sort the task set.

Step 3. Select task i as a candidate.

Step 4. Set the number of time windows of task i on all satellites as $|W_i|$. If $|W_i| = 0$, then go to Step 7; else randomly sort the time windows of task i. Set $k = 1$.

Step 5. Select W_{ik} as a candidate; if W_{ik} satisfies constraints (6)–(8), then insert it into I_j, update the start time and finish time of task i in I_j, and then go to Step 7.

Step 6. Let $k = k + 1$; if $k \leq |W_i|$, then go to Step 5.

Step 7. Let $i = i + 1$; if $i > n_t$, turn to Step 8; else, turn to Step 3.

Step 8. Let $j = j + 1$; if $j \leq M$, then set $i = 1$ and go to Step 2; else end the initialization.

(3) Selection. We use the tournament selection operator to select two individuals from the population. Next, the individual that dominates the other is selected.

(4) Crossover. We use single-point crossover as a means to mate the parent chromosomes. Single-point crossover selects a locus in the two parent chromosomes, which is called the crossover point. In the paper, we select the first gene on the second chromosome as a crossover point. After this point is chosen, the string of genes prior to the crossover point in parent chromosome 1 and parent chromosome 2 is transposed. If there are two identical genes on the offspring chromosome 1 and the offspring chromosome 2, then the gene that is located on the offspring chromosome 1 is deleted. Compared to the parent chromosome, some of the missing genes are inserted into the offspring chromosome. Figure 6 shows an example of the crossover.

(5) Mutation. According to the property that a task may have multiple available opportunities, we design the single-point mutation operator. In the algorithm, the mutation probability is set to 0.01. The process of the mutation operation on a chromosome is described as follows.

Step 1. Sort the sequence of tasks in task set T randomly and then take the serial number i of the task in the queue in order.

Step 2. Compute the number of available opportunities $|W_i|$ for task i.

Step 3. If $|W_i| = 0$, then $i = i + 1$ and go to Step 2.

Step 4. If there exists task i on the chromosome, then check if it can be rearranged to another position. If it can be moved, then rearrange it to the other position; else remove it from the chromosome. End.

Step 5. If task i is not located on a chromosome, attempt to insert the task into the chromosome. If successful, then end; else $i = i + 1$ and go to Step 2.

Finally, the process of the multiobjective genetic algorithm is given as follows [29].

Step 1. Set the parameters of the population size M, the maximum number of iterations T, and the current number of iteration times $t = 0$. Design the individual genetic code of the solution. Randomly generate initial population P_t.

Step 2. Conduct the selection, crossover, and mutation operators on population P_t to generate new populations Q_t. Set $P_{t+1} = \emptyset$, and set the counter $i = 1$.

Step 3. Conduct nondominated sorting on populations $R_t = P_t \cup Q_t$ and obtain the Pareto front $F = (F_1, F_2 \ldots)$ of R_t.

Step 4. Let $P_{t+1} = P_{t+1} \cup F_i$, $i = i + 1$.

Step 5. If $|P_{t+1}| + |F_i| < M$, then go to Step 4; if $|P_{t+1}| + |F_i| = M$, then go to Step 7.

Step 6. Calculate the crowding distance of individuals in F_i. Next, select noninferior individuals according to the ascending order of the crowding distances. The number of selected individuals is $|M - |P_{t+1}||$, where $P_{t+1} = P_{t+1} \cup F_i[1 : |M - |P_{t+1}||]$.

Step 7. Let $t = t + 1$; if $t < T$ and $P_t \cap P_{t-1} \neq P_{t-1}$, then go to Step 2.

Step 8. Output the nondominated Pareto solutions and then decode them.

Using the multiobjective genetic algorithm NSGA-II, we obtain feasible initial schedules. Each schedule is the sequence of tasks ordered in time for the EOSs.

4. Dynamic Adjustment

The initial schedule, which is produced in the first stage, needs uploading. The interval time between schedule building and uploading depends on the practical needs. Suppose a number of emergency tasks arrive after the schedule uploading. In this case, the initial schedule must be adjusted. A new schedule SS′ is made by adjusting the initial scheduling scheme dynamically with the arrival of new emergency tasks. The process of adjustment is the second stage of satellite

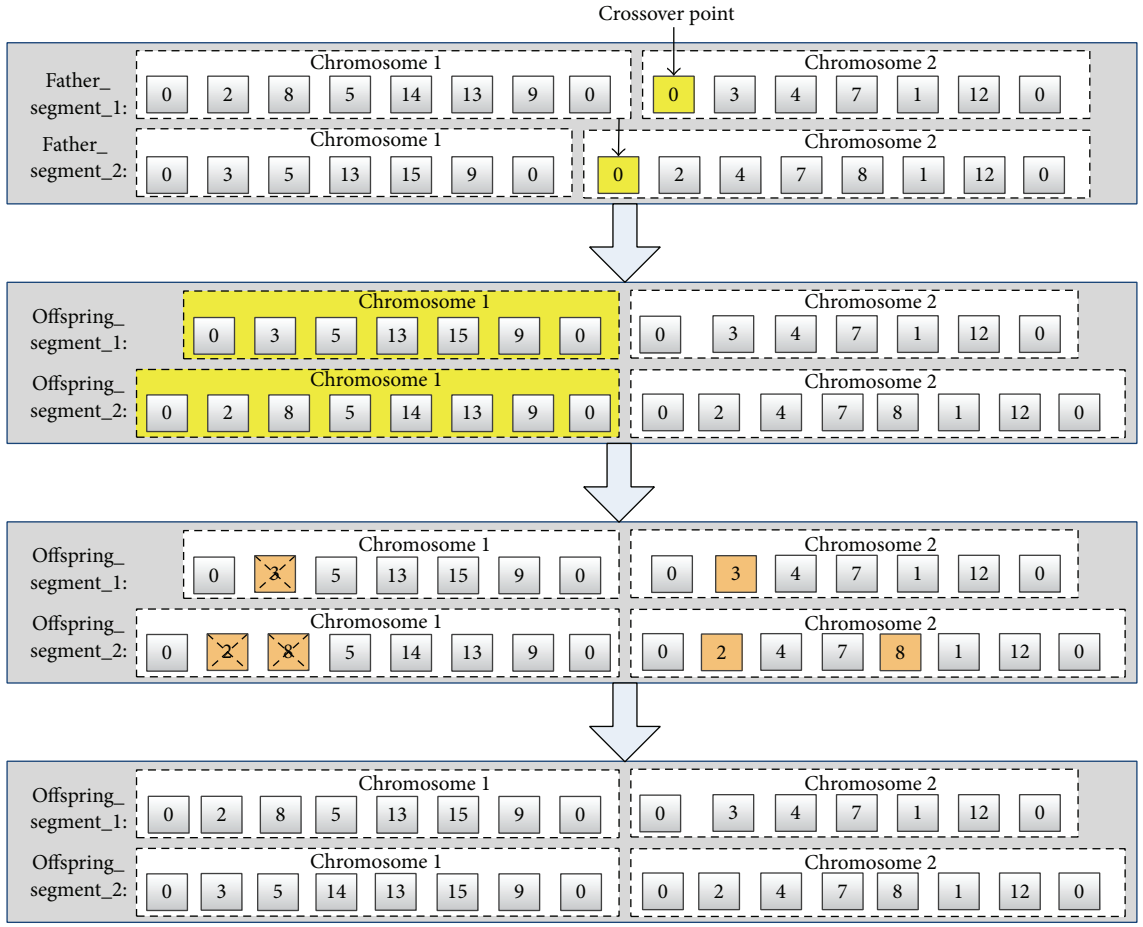

Figure 6: Crossover operator.

dynamic scheduling. When dynamically adjusting the initial schedule, the solution stability is an important problem. Although we can completely reschedule the initial scheme to obtain the optimal income, it may result in the phenomenon of "shock" for the scheme. To minimize the disturbance to the initial scheme while maintaining high revenue, we designed a rule-based heuristic algorithm. For the purpose of increasing the number of scheduled tasks, a novel compact task merging method considering the duration of task execution is proposed.

4.1. Rule-Based Heuristic Algorithm. In the stage of dynamic adjustment, with the arrival of new emergency tasks $T^E = \{t_{N+1}, t_{N+2}, \ldots, t_{N+N^E}\}$, we expect that the optimal solution satisfies two objectives: (1) maximum revenue

$$\max : R\left(\mathrm{SS}^*\right) = \sum_{i=1}^{N+N^E} \sum_{j=1}^{M} \sum_{k=1}^{K_{ij}} x_{i,k}^{j} p_i \tag{12}$$

and (2) minimum difference between the adjusted schedule SS^* and the initial schedule SS

$$\max : \left|\mathrm{SS}^* \cap \mathrm{SS}\right| \tag{13}$$

while satisfying the four constraints which are represented by (5)–(8).

When assigning dynamic tasks to a schedule, besides high revenue, schedule stability is very important. In practice, satellite scheduling is a complicated process. A certain amount of time is required to upload an instruction to a satellite with special equipment within limited visible time windows [19, 20]. Once the scheduling scheme is modified on a large scale, there may be a series of influences on the decision of users. Although higher revenue may be obtained by complete rescheduling, the excessive changes may cause substantial operational issues. In addition, such excessive changes may lead to a large-scale adjustment for the scheduling scheme, which results in the phenomenon of "shock" for the scheduling scheme.

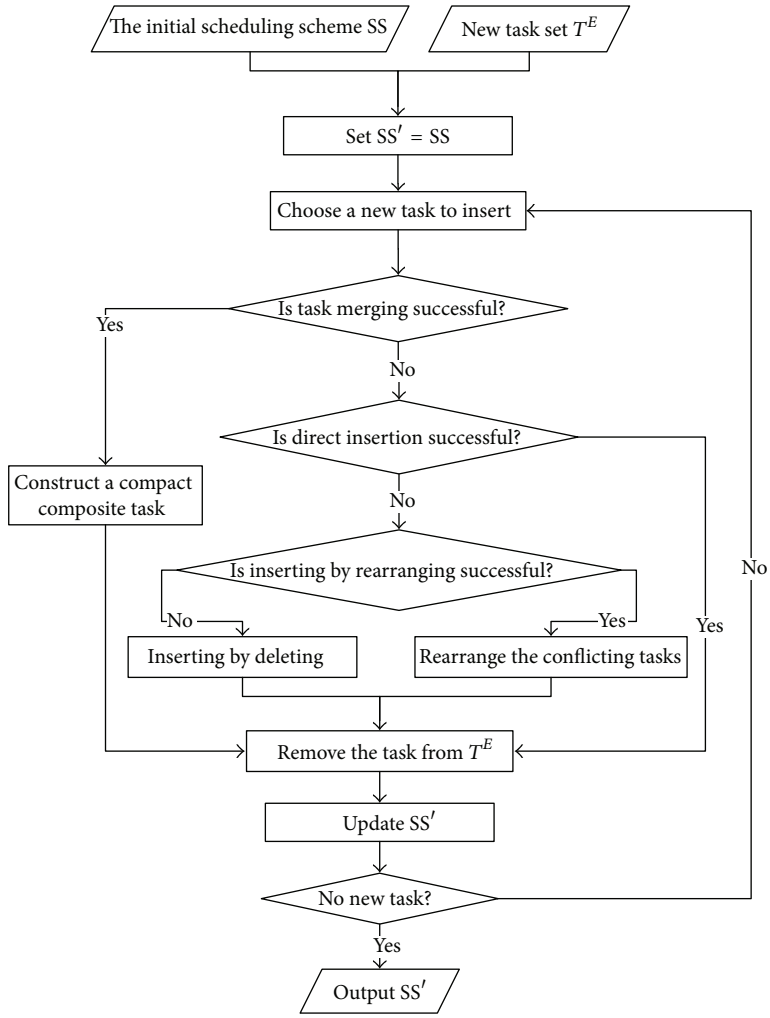

Figure 7: The process of inserting a new task.

The two objectives of the algorithm are incompatible to some extent. That is, the adjusted schedule that has the maximum revenue may differ greatly from the initial one. To achieve satisfactory solutions, the rule-based heuristic algorithms are feasible algorithms. Therefore, in this paper, we design a dynamic adjusting rule-based heuristic algorithm considering compact task merging (CTM-DAHA). To accommodate the new tasks into the initial schedule, four methods are used: task merging, direct insertion, insertion by rearranging, and insertion by deleting. Figure 7 shows for a given new task how to insert it into the schedule.

The design of the heuristic rules of CTM-DAHA is as follows.

Heuristic rule 1: select a new task from T^E to insert according to the needs of the tasks from high to low. The need of an observation task indicates how badly the task needs to be performed [30]. The need is defined as

$$\text{Need}\left(t_i\right) = \frac{p_i}{\left|\text{AO}_i\right|}. \tag{14}$$

Heuristic rule 2: if there is more than one time window of a new task, then select time windows to perform the task according to the contention, ranked from low to high.

For a given time window w, we could measure the contention by counting the number of tasks that need that time window, weighted by the weights of the tasks:

$$\text{contention}\left(w\right) = \sum_{t_i \in T(w)} p\left(t_i\right), \tag{15}$$

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

Heuristic rule 3: for a given task, if its multiple time windows W have the same contention, then select the time window according to the associated slewing angle, ranked from small to large.

Heuristic rule 4: for a given task, if its multiple time windows W have the same contention and the associated

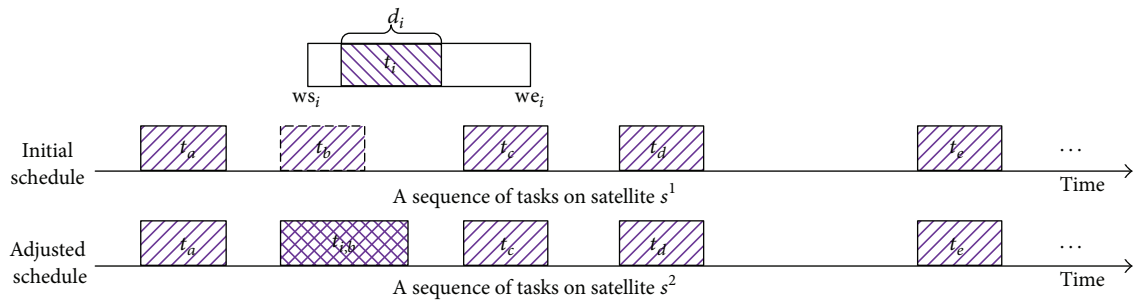

Figure 8: Inserting a new task into initial schedule by task merging.

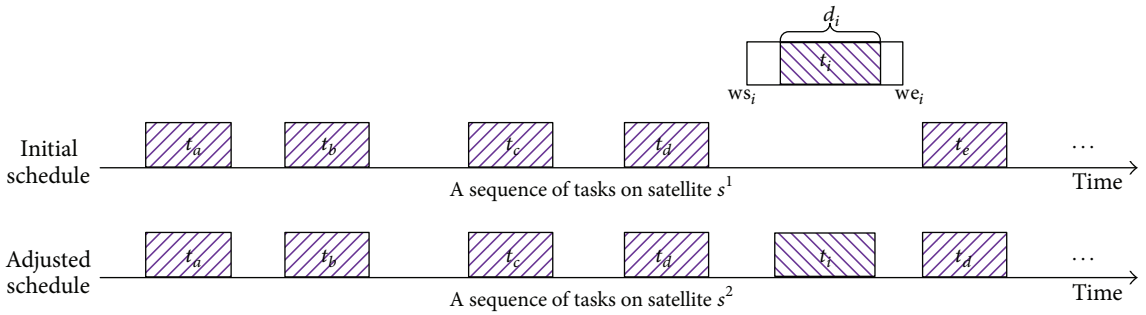

Figure 9: Inserting a new task directly into the initial scheme.

slewing angles are equal, then select the time window that has the earliest start time.

Heuristic rule 5: for a given task associated with a time window, there are four methods that can be used to schedule the task into the initial schedule: (1) insertion by task merging; (2) direct insertion; (3) insertion by rearranging; (4) insertion by deleting.

(1) Insertion by Task Merging. For a given task, we first determine if it can merge with any other existing task in the schedule. As shown in Figure 8, new tasks t_i and t_b can be constructed as a compact composite task $t_{i,b}$ on satellite s^1.

(2) Direct Insertion. When task merging fails, the task will be inserted into a free timeslot directly if it does not conflict with any other task in the time window. As shown in Figure 9, task t_i can be inserted directly.

(3) Insertion by Rearranging. When inserting a new task t_i directly fails, t_i conflicts with one or more scheduled tasks in a specific time window w. The set T_i^C is defined as the conflicting tasks of task i. If each task $t_j \in T_i^C$ can be rearranged in another timeslot, then the task t_i can be inserted into the time window w. To rearrange conflicting tasks, we employ the repair search method [18]. As shown in Figure 10, task t_i conflicts with scheduled task t_c on s^1. Task t_c is reassigned on another timeslot on s^2 and task t_i is inserted on s^1 in the schedule.

(4) Insertion by Deleting. When inserting a new task t_i by rearranging fails, if the property of t_i is larger than the total properties of the conflicting tasks, then the new task is assigned while the conflicting tasks are removed from the schedule. As shown in Figure 11, t_c is removed from the schedule and t_i is assigned.

The procedure of the rule-based heuristic algorithm is described as in Algorithm 1.

4.2. Compact Task Merging Method. If two or more targets are geographically adjacent, we can rationally tune the slewing angle and the observation duration of the sensor to enable an observation strip to cover them. In other words, the tasks in the same swath of a sensor may be merged into one composite task. For the purpose of improving the imaging opportunities of the new tasks, a task merging strategy is required. By considering the duration of task execution, we employ a new compact task merging method to construct the so-called 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.

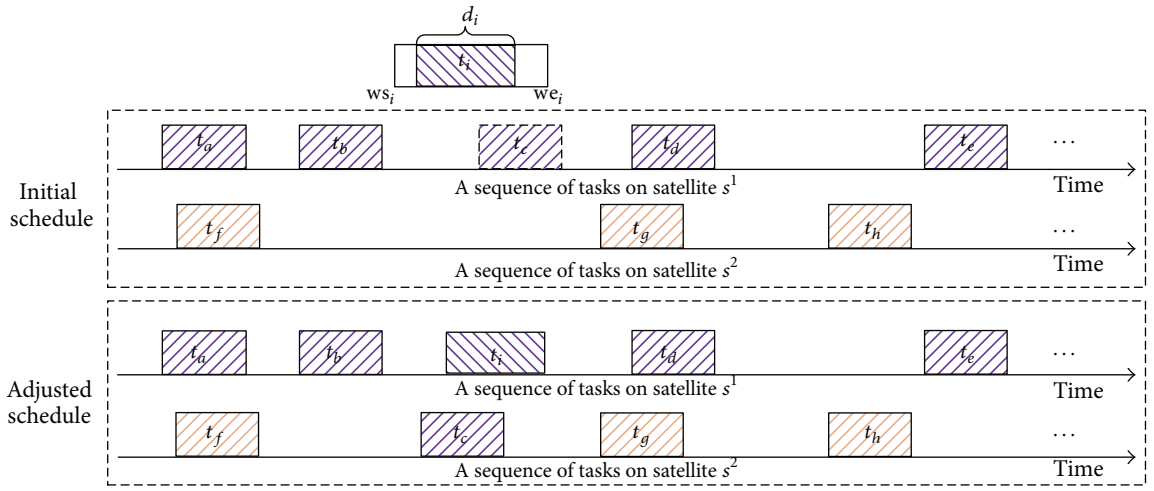

FIGURE 10: Inserting a new task into the initial scheme by rearranging the conflicting tasks.

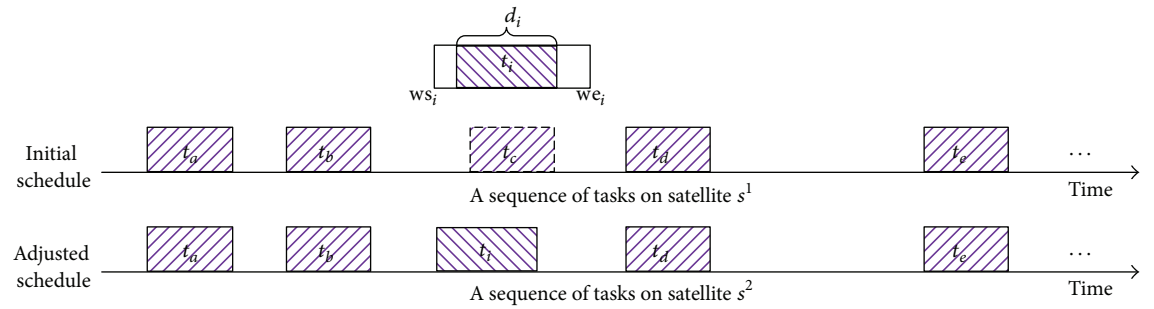

FIGURE 11: Inserting a new task into initial scheme by deleting conflicting tasks.

Suppose there are two tasks t_i and t_j that could be imaged by satellite j. And $\text{ao}_i = \{[\text{ws}_i, \text{we}_i], \theta_i\} \in \text{AO}_i^j$ is an available opportunity of task t_i. Accordingly, $\text{ao}_j = \{[\text{ws}_j, \text{we}_j], \theta_j\} \in \text{AO}_j^j$ is an available opportunity of task t_j.

According to the traditional merging idea, two tasks t_i and t_j can be combined into a composite task $t_{i,j}$ if and only if the following condition must hold [12, 22–25]:

$$\begin{aligned} \max\left\{\text{we}_i, \text{we}_j\right\} - \min\left\{\text{ws}_i, \text{ws}_j\right\} &\le \Delta d, \\ \left|\theta_i - \theta_j\right| &\le \Delta\theta, \end{aligned} \tag{16}$$

where Δd and $\Delta\theta$ are the longest open time and field of view of sensor on satellite j, respectively. The time window and slewing angle of composite task $t_{i,j}$ are calculated as

$$\begin{aligned} W_{i,j} &= \left[\min\left\{\text{ws}_i, \text{ws}_j\right\}, \max\left\{\text{we}_i, \text{we}_j\right\}\right], \\ \theta_{i,j} &= \frac{\theta_i + \theta_j}{2}. \end{aligned} \tag{17}$$

We call these tasks metatasks if they can be combined into one composite task. The composite task obtained by the above task merging method is traditionally characterized as the union of visible time windows and mean of slewing angles of its metatasks. 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.

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 not later than that of task t_j in the following.

Theorem 7. *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*

$$\left(ws_j + d_j\right) - \left(we_i - d_i\right) \le \Delta d, \tag{18}$$

$$\left|\theta_i - \theta_j\right| \le \Delta\theta. \tag{19}$$

Equation (18) is time window constraint. As shown in Figure 12(b), we illustrate the case that two time windows of tasks have intersection. In fact, three types of temporal relationships exist between the time windows of two tasks, that is, disjoint, intersected, and containing. The details are presented

(1) sort the new emergency task set T^E according to the needs of the tasks
(2) for all $i \in [N+1, \ldots, N+N^E]$, do
(3) for all $j \in [1, \ldots, M]$, 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) while $T^E \neq \varnothing$ do
(9) set $SS' = SS$
(10) take the serial number i of the first task in the queue
(11) if $|W_i| > 1$ do
(12) calculate contention(W_i)
(13) sort the set W_i based on Heuristic rule 2, rule 3, and rule 4
(14) end if
(15) for $k \in [1, 2, \ldots, |W_i|]$ do
(16) for all scheduled tasks in SS' do
(17) if w_{ik} can be merged with the time window of a scheduled task t_j
(18) construct a compact composite task t_{ij} between t_i and t_j
(19) remove task i from T^E
(20) goto step (8)
(21) end if
(22) end for
(23) end for
(24) for $k \in [1, 2, \ldots, |W_i|]$ do
(25) if w_{ik} can be inserted into SS' directly do
(26) insert t_i and determine the earliest start time ES_i^j and the latest start time LS_i^j of t_i
(27) remove task i from T^E
(28) goto step (8)
(29) end if
(30) end for
(31) for $k \in [1, 2, \ldots, |W_i|]$ do
(32) for all scheduled tasks in SS' do
(33) if w_{ik} can be inserted into SS' by rearranging the conflict-tasks
(34) rearrange the conflict-tasks in SS'
(35) insert t_i and determine the earliest start time ES_i^j and the latest start time LS_i^j of t_i
(36) remove task i from T^E
(37) goto step (8)
(38) end if
(39) else
(40) if the weight of t_i is larger than the total priorities of conflict-tasks
(41) remove the conflict-tasks of t_i from SS'
(42) insert t_i and determine the earliest start time ES_i^j and the latest start time LS_i^j of t_i
(43) remove task i from T^E
(44) goto step (8)
(45) end if
(46) else
(47) remove task i from T^E
(48) goto step (8)
(49) end for
(50) end for
(51) end while

Algorithm 1

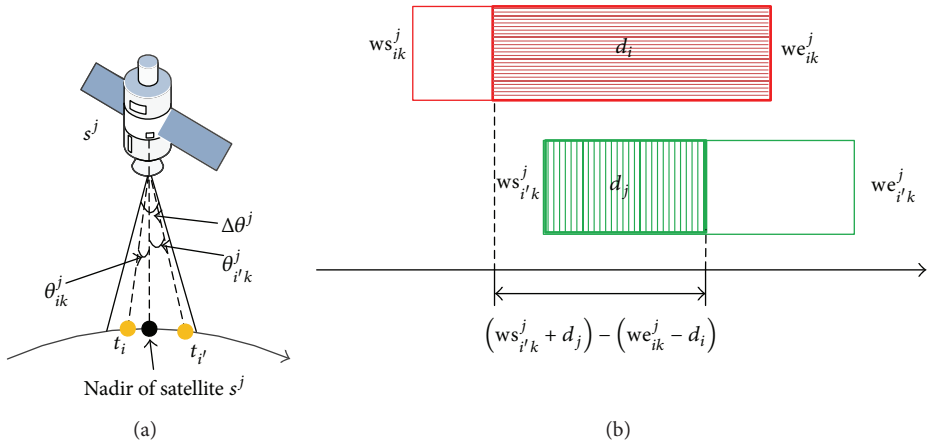

Figure 12: Task merging constraints.

in Section 2 in Supplementary File (see Supplementary Material available online at http://dx.doi.org/10.1155/2015/731734).

Equation (19) is slewing angle constraint. It is shown in Figure 12(a) that tasks t_i and t_j must be located in the same swath of the sensor.

Theorem 8. *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*

$$ws_{i,j} = \begin{cases} we_i - d_i, & if \ \left|W_i \cap W_j\right| \le \min\left(d_i, d_j\right) \\ \min\left\{ws_j, \max\left(ws_i, ws_j + d_j - d_i\right)\right\}, & else \end{cases} \tag{20}$$

to

$$we_{i,j} = \begin{cases} w_j + d_j, & if \ \left|W_i \cap W_j\right| \le \min\left(d_i, d_j\right) \\ \min\left(we_i - d_i, we_j - d_j\right) + \max\left(d_i, d_j\right), & else \end{cases} \tag{21}$$

and its indispensable duration of task execution should be

$$d_{i,j} = \begin{cases} ws_j + d_j - \left(we_i - d_i\right), & if \ \left|W_i \cap W_j\right| \le \min\left(d_i, d_j\right) \\ \max\left(d_i, d_j\right), & else \end{cases} \tag{22}$$

and the slewing angle is given by

$$\theta_{i,j} = \begin{cases} \max\left\{\theta_i - \dfrac{\Delta\theta_s}{2}, 0\right\}, & if \ \theta_i \ge 0, \ \left|\theta_i\right| \ge \left|\theta_j\right| \\ \min\left\{\theta_i + \dfrac{\Delta\theta_s}{2}, 0\right\}, & if \ \theta_i < 0, \ \left|\theta_i\right| \ge \left|\theta_j\right| \\ \max\left\{\theta_j - \dfrac{\Delta\theta_s}{2}, 0\right\}, & if \ \theta_j \ge 0, \ \left|\theta_j\right| > \left|\theta_i\right| \\ \min\left\{\theta_j + \dfrac{\Delta\theta_s}{2}, 0\right\}, & if \ \theta_j < 0, \ \left|\theta_j\right| > \left|\theta_i\right|. \end{cases} \tag{23}$$

Proof. Please refer to Sections 2 and 3 of the Supplementary File for the detailed proof. □

5. Experimental Simulation and Discussion

In this section, we test our model and algorithm by experimental simulations. For convenience, the robust satellite scheduling model with three objectives proposed in our paper is called RSSM3, while the robust satellite scheduling model without the objective of total task execution time is called RSSM2.

RSSM3 is the robust satellite scheduling model that consists of three objectives: the maximum revenue $R(\text{SS})$, the maximum value of the neighborhood-based robust indicator $N(\text{SS})$, and the minimum total duration of task execution $T(\text{SS})$.

Table 1: Parameters of satellites.

Parameters	s^1	s^2
Maximum slewing angle/°	32	32
Field of view/°	2.203	2.203
Eccentricity	0.0001879	0.0006165
Inclination/°	98.4745	97.4097
Argument of perigee/°	76.4542	72.083
RAAN/°	145.131	145.956
Mean anomaly/°	283.684	41.4996
Mean motion revs/day	14.354	15.2108

RSSM2 is a robust satellite scheduling model that consists of two objectives: $R(\mathrm{SS})$ and $N(\mathrm{SS})$.

In the first stage, we use RSSM3 and RSSM2 to generate two sets of feasible solutions. Furthermore, we select the solution of RSSM2 as the initial scheduling scheme SS2 and one of the feasible solutions of RSSM3 as the initial schedule SS3. In the second stage, to demonstrate the advantage of total execution time, SS2 and SS3 are adjusted using CTM-DAHA. To evaluate the advantage of compact task merging method, we use CTM-DAHA and TTM-DAHA, respectively, to adjust SS3.

TTM-DAHA is dynamic adjusting heuristic algorithm considering traditional task merging. Compared to CTM-DAHA, the algorithm uses traditional task merging strategy to construct composite tasks.

5.1. The Design of Experiment. To evaluate performance of the proposed method, we simulate two satellites to accomplish imaging tasks. Each satellite circles the earth approximately 100 min each time. Each satellite loads one sensor which can slew laterally over the angular range of $[-32^\circ, 32^\circ]$. Some parameters of the two satellites are presented in Table 1. The scheduling horizon is 24 h. In addition, we assume the maximum time for each satellite opening its sensor once is 60 seconds and the longest imaging time in any period time of orb^j is 150 seconds.

We simulate 200 common tasks which are randomly generated in the surface of the earth. Three group emergency tasks are given different size: 20, 40, and 60. Without loss of generality, the priorities of all tasks are randomly distributed in [1, 10]. The weight and duration of each task are shown in the appendix of the paper.

The scheduling period considered is 7 March 2014 12:00:00.000 UTCG~8 March 2014 12:00:00.000 UTCG. Before scheduling we compute the time window and slewing angle for each task using STK (Satellite Tool Kit). The time window is removed if its span is shorter than the duration of the corresponding task.

5.2. The Simulation Result

5.2.1. The Initial Scheduling Scheme. The feasible solutions that are obtained by RSSM3 and RSSM2 are shown in Figure 13.

We describe all the solutions of RSSM2 and RSSM3 in two-dimensional space and three-dimensional space, respectively. From Figure 13, RSSM2 is found to generate only one solution, while RSSM3 is found to produce multiple solutions because of the added objective of total task execution duration. In addition, the solution of RSSM2 that has the highest revenue may not be the best because it requires the longest duration of task execution.

We select the solution ($R(\mathrm{SS2}) = 1123$; $N(\mathrm{SS2}) = 923$; $T(\mathrm{SS2}) = 2808$) of RSSM2 as initial scheduling scheme SS2 and the solution ($R(\mathrm{SS3}) = 1122$; $N(\mathrm{SS3}) = 898$; $T(\mathrm{SS3}) = 2795$) from the solutions of RSSM3 as the initial scheduling scheme SS3.

5.2.2. The Adjusted Scheduling Scheme. In the second stage, with the arrival of the three group emergency tasks, we use CTM-DAHA to adjust SS2 and SS3 so as to obtain new schedules $\mathrm{SS2}^*_{\text{C-20}}$, $\mathrm{SS2}^*_{\text{C-40}}$, $\mathrm{SS2}^*_{\text{C-60}}$ and $\mathrm{SS3}^*_{\text{C-20}}$, $\mathrm{SS3}^*_{\text{C-40}}$, and $\mathrm{SS3}^*_{\text{C-40}}$, respectively. The new schedules $\mathrm{SS3}^*_{\text{T-20}}$, $\mathrm{SS3}^*_{\text{T-40}}$, and $\mathrm{SS3}^*_{\text{T-60}}$ are produced through adjustment of SS3 by TTM-DATA.

The simulation results are presented in Table 2.

5.3. Discussion. In the first stage, we obtain different solutions using RSSM3 and RSSM2, respectively. As shown in Figure 13, there is only one feasible solution generated by RSSM2 while RSSM3 can provide multiple solutions.

When there are 20 emergency tasks, $\mathrm{SS3}^*_{\text{C-20}}$ produces the higher revenue compared with $\mathrm{SS2}^*_{\text{C-20}}$. In addition, the disturbance of $\mathrm{SS3}^*_{\text{C-20}}$ is lower than that of $\mathrm{SS2}^*_{\text{C-20}}$. As a result, RSSM3 can produce a schedule with high robustness as well as large revenue. The result proves that the objective of total task execution time is a useful robust indicator. Comparing $\mathrm{SS3}^*_{\text{C-20}}$ with $\mathrm{SS3}^*_{\text{T-20}}$, $\mathrm{SS3}^*_{\text{C-20}}$ has the larger revenue and more numbers of merging tasks with shorter imaging time and a smaller slewing angle. This result indicates that more tasks

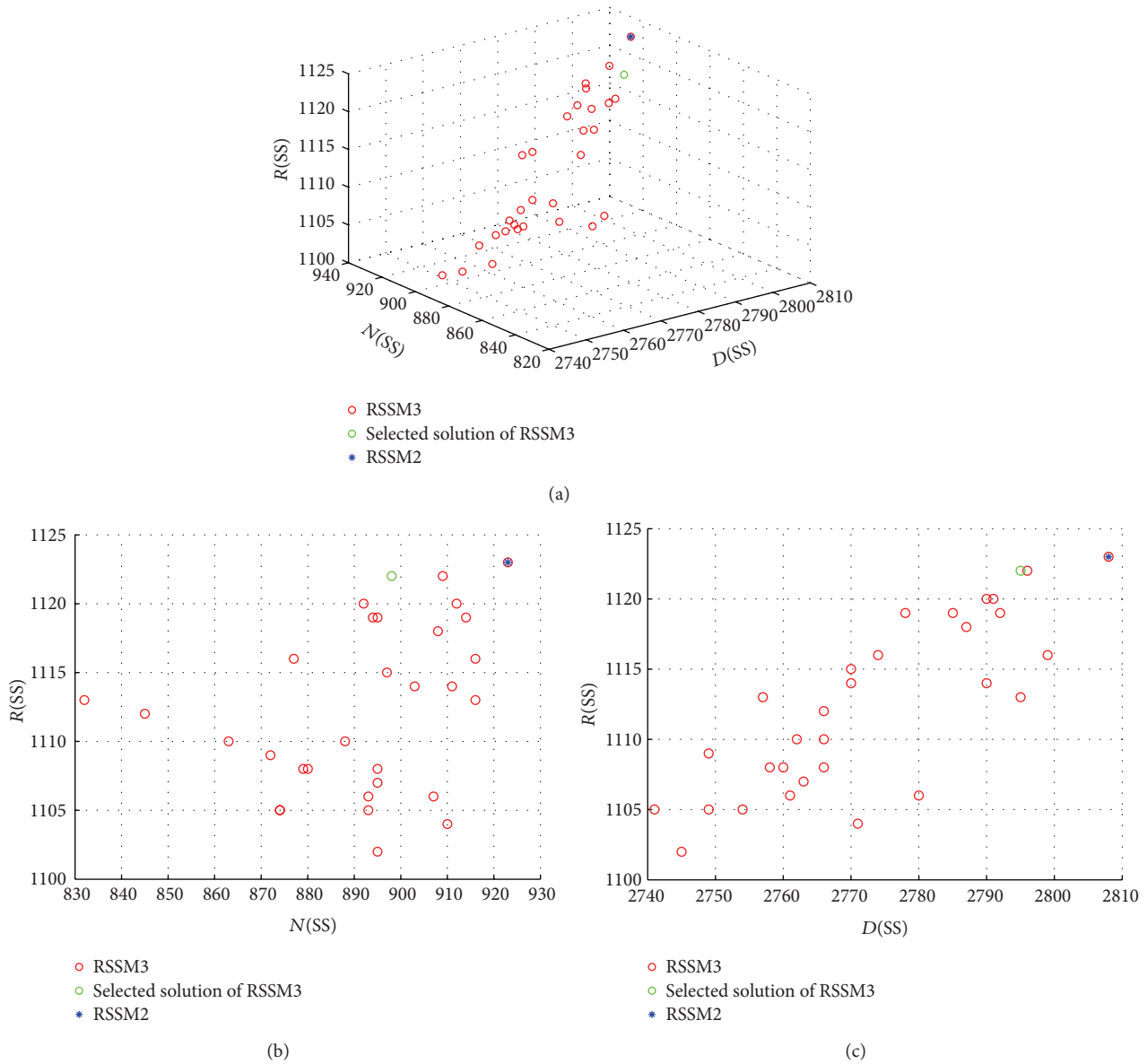

Figure 13: The values of objectives of RSSM2 and RSSM3.

Table 2: The results of dynamic adjustment.

N	Schedule	Total revenue	Disturbance	NC	NITM	NI	NIR	NID
20	$SS2^*_{C\text{-}20}$	1263	6	20	6	9	4	1
	$SS3^*_{C\text{-}20}$	1264	3	20	6	11	2	1
	$SS3^*_{T\text{-}20}$	1250	3	18	4	11	2	1
40	$SS2^*_{C\text{-}40}$	1361	10	39	6	24	7	2
	$SS3^*_{C\text{-}40}$	1368	4	39	6	29	3	1
	$SS3^*_{T\text{-}40}$	1345	6	37	4	27	4	2
60	$SS2^*_{T\text{-}60}$	1466	16	56	9	32	11	4
	$SS3^*_{C\text{-}60}$	1480	7	58	10	41	4	3
	$SS3^*_{T\text{-}60}$	1462	8	55	4	43	4	4

Notes: N-the number of new tasks, disturbance-the number of common tasks that are rearranged in the adjusted schedule, NC: the number of emergency tasks completed, 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.

Table 3: The duration and properties of common tasks.

Task	Weight	Duration/s	Task	Weight	Duration/s
1	2	17	101	9	7
2	9	6	102	4	23
3	5	21	103	6	5
4	6	23	104	9	20
5	9	18	105	6	21
6	10	19	106	4	22
7	3	19	107	8	7
8	3	12	108	3	13
9	3	17	109	2	10
10	1	8	110	7	20
11	3	18	111	1	13
12	8	6	112	5	22
13	9	10	113	5	8
14	10	6	114	4	10
15	4	7	115	6	8
16	8	21	116	5	8
17	5	18	117	10	22
18	2	11	118	5	16
19	6	23	119	4	15
20	2	6	120	7	8
21	4	13	121	8	21
22	9	12	122	3	17
23	1	20	123	5	12
24	9	20	124	3	15
25	4	9	125	5	13
26	5	14	126	9	6
27	10	13	127	1	10
28	1	17	128	3	7
29	4	18	129	4	8
30	8	19	130	6	10
31	4	10	131	5	13
32	2	18	132	10	6
33	5	17	133	7	22
34	3	8	134	3	23
35	7	7	135	9	14
36	6	14	136	10	14
37	4	23	137	7	11
38	5	11	138	1	22
39	3	16	139	10	12
40	9	9	140	5	7
41	1	19	141	10	20
42	1	10	142	3	12
43	5	15	143	7	10
44	7	18	144	8	13
45	1	22	145	8	7
46	4	23	146	9	8
47	5	15	147	6	23
48	3	8	148	8	23
49	4	8	149	4	16
50	4	10	150	8	6
51	1	21	151	7	9
52	5	10	152	1	12
53	5	20	153	5	21
54	10	10	154	8	5
55	7	23	155	5	6
56	8	12	156	6	8
57	9	9	157	10	17
58	3	10	158	9	19
59	9	17	159	8	17
60	8	14	160	2	14
61	5	12	161	5	15
62	9	21	162	7	11
63	7	16	163	6	19
64	5	15	164	3	9
65	7	22	165	9	18
66	9	10	166	1	8
67	9	19	167	6	12
68	10	19	168	4	17
69	10	12	169	8	20
70	6	16	170	4	7
71	4	6	171	4	23
72	1	6	172	6	20
73	9	15	173	3	14
74	7	20	174	1	13
75	6	23	175	8	13
76	1	7	176	4	11
77	9	16	177	8	15
78	2	14	178	7	15
79	9	5	179	3	21
80	8	11	180	3	20
81	3	8	181	8	17
82	3	20	182	5	12
83	9	11	183	4	20
84	7	15	184	7	15
85	4	8	185	8	12
86	1	16	186	8	23
87	4	10	187	9	22
88	8	17	188	5	15
89	4	18	189	9	17
90	10	19	190	4	16
91	1	14	191	9	9
92	9	7	192	5	11
93	8	9	193	2	14
94	8	22	194	2	9
95	8	8	195	5	21
96	1	21	196	6	9
97	1	15	197	10	9
98	5	24	198	10	13
99	4	6	199	1	1
100	1	13	200	1	1

Table 4: The duration and properties of new emergency tasks.

Task	Weight	Duration/s	Task	Weight	Duration/s
201	8	14	231	5	13
202	7	15	232	6	12
203	7	13	233	9	13
204	9	6	234	2	18
205	9	11	235	4	11
206	4	22	236	3	10
207	7	20	237	1	23
208	8	17	238	5	6
209	6	15	239	6	5
210	9	21	240	9	13
211	3	20	241	5	8
212	9	18	242	9	15
213	7	13	243	8	12
214	10	12	244	4	8
215	5	7	245	9	14
216	10	12	246	5	5
217	7	12	247	6	10
218	6	11	248	5	12
219	5	9	249	9	10
220	7	8	250	4	10
221	9	16	251	2	14
222	6	21	252	7	9
223	7	7	253	4	23
224	7	19	254	9	17
225	6	9	255	8	10
226	7	7	256	9	5
227	7	21	257	2	11
228	4	23	258	5	14
229	3	8	259	3	18
230	4	12	260	9	19

Table 5: The adjusted common tasks in $SS2^{*}_{C\text{-}20}$.

Task	Satellite	The earliest start time	The latest start time	Duration/s	Slewing angle/°	The way to adjust tasks
97	—	—	—	—	—	D
193	—	—	—	—	—	D
169	1	19825	19846	20	0	R
14	1	32626	32661	6	0	R
120	1	43300	43333	8	0	R
35	1	54447	54481	7	0	R

Table 6: The inserted emergency tasks in $SS2^{*}_{C\text{-}20}$.

Task	Satellite	The earliest start time	The latest start time	Duration/s	Slewing angle/°	The way to insert tasks
206	1	3240	3259	22	20	IR
207 and 92	1	7327	7335	20	1.797	TM
208	1	9926	9950	17	15	IR
150 and 210	1	12550	12570	21	0.797	TM
219	1	13854	13886	9	22	IR
46 and 204	1	26593	26606	21	0	TM
205	1	22177	22208	11	23	ID
220	1	30146	30180	8	3	I
213	1	34540	34570	13	9	I
214	1	57054	57083	12	2	I
209	1	58682	58709	15	10	IR
202	1	60137	60164	15	3	I
217	1	63500	63529	12	29	I
215	1	74614	74649	7	6	IR
197 and 211	2	5493	5493	33	0.797	TM
201	2	14634	14647	14	2	I
216	2	17246	17271	12	5	I
212 and 179	2	22815	22815	21	0	TM
203 and 95	2	52998	53021	23	0.797	TM
218	2	69591	69628	37	0.797	I

Table 7: The adjusted common tasks in $SS3^{*}_{C\text{-}20}$.

Task	Satellite	The earliest start time	The latest start time	Duration/s	Slewing angle/°	The way to adjust tasks
97	—	—	—	—	—	D
169	1	19825	19846	20	0	R
33	2	63000	63008	17	0	R

Table 8: The inserted emergency tasks in $SS3^{*}_{C\text{-}20}$.

Task	Satellite	The earliest start time	The latest start time	Duration/s	Slewing angle/°	The way to insert tasks
206	1	3240	3259	22	20	I
207 and 92	1	7327	7335	20	1.797	TM
208	1	9926	9950	17	15	I
150 and 210	1	12550	12570	21	0.797	TM
219	1	13854	13886	9	22	IR
205	1	22177	22208	11	23	ID
204	1	26592	26606	6	1	I
220 and 166	1	30146	30175	8	0.797	TM
213	1	34540	34570	13	9	I
214	1	57054	57083	12	2	I
209	1	58682	58709	15	10	I
202	1	60137	60164	15	3	I
217	1	63500	63529	12	29	I
215	1	74614	74649	7	6	IR
197 and 211	2	5493	5493	33	0.797	TM
201	2	14634	14647	14	2	I
216	2	17246	17259	12	5	I
212 and 179	2	22815	22815	21	0	TM
203 and 95	2	52998	53005	13	0.797	TM
218	2	69588	69602	11	3	I

Table 9: The adjusted common tasks in $SS3^{*}_{T\text{-}20}$.

Task	Satellite	The earliest start time	The latest start time	Duration/s	Slewing angle/°	The way to adjust tasks
97	—	—	—	—	—	D
169	1	19825	19846	20	0	R
33	2	63000	63008	17	0	R

Table 10: The inserted emergency tasks in $SS3^{*}_{T\text{-}20}$.

Task	Satellite	The earliest start time	The latest start time	Duration/s	Slewing angle/°	The way to insert tasks
206	1	3240	3259	22	20	I
208	1	9926	9950	17	15	I
210 and 150	1	12550	12550	53	1.5	TM
219	1	13854	13886	9	22	IR
205	1	22177	22208	11	23	ID
204	1	26570	26606	6	1	I
220 and 166	1	30141	30141	47	1.5	TM
213	1	34540	34570	13	9	I
214	1	57054	57083	12	2	I
209	1	58682	58709	15	10	I
202	1	60137	60164	15	3	I
217	1	63500	63529	12	29	I
215	1	74614	74649	7	6	IR
211 and 197	2	5488	5488	54	1.5	TM
201	2	14634	14647	14	2	I
216	2	17246	17259	12	5	I
212 and 179	2	22808	22808	32	1	TM
218	2	69588	69602	11	3	I

Table 11: The adjusted common tasks in $SS2^{*}_{C\text{-}40}$.

Task	Satellite	The earliest start time	The latest start time	Duration/s	Slewing angle/°	The way to adjust tasks
97	—	—	—	—	—	D
193	—	—	—	—	—	D
70	—	—	—	—	—	D
169	1	19825	19846	20	0	R
14	1	32626	32661	6	0	R
120	1	43300	43333	8	0	R
35	1	54447	54481	7	0	R
28	1	61374	61398	17	0	R
143	2	84955	84986	10	0	R
134	2	54003	54006	23	0	R

Table 12: The inserted emergency tasks in $SS2^{*}_{C\text{-}40}$.

Task	Satellite	The earliest start time	The latest start time	Duration/s	Slewing angle/°	The way to insert tasks
233	1	1282	1320	13	4	I
206	1	3240	3259	22	20	IR
207 and 92	1	7327	7335	20	1.797	TM
208	1	9926	9950	17	15	IR
238	1	11625	11661	6	30	I
223	1	12308	12342	7	15	IR
150 and 210	1	12550	12570	21	0.797	TM
236	1	13088	13120	10	28	I
231	1	14938	14966	13	20	IR
205	1	22177	22208	11	23	ID
228	1	24349	24367	23	25	I
229	1	25154	25187	8	3	I
46 and 204	1	26593	26606	23	0	TM
220	1	30146	30180	8	3	I
213	1	34540	34570	13	9	I
214	1	57054	57083	12	2	I
221	1	58115	58141	16	29	I
227	1	58362	58382	21	6	I
209	1	58682	586709	15	10	IR
202	1	60137	60164	15	3	I
217	1	63500	63529	12	29	I
230	1	64352	64382	12	4	I
215	1	74614	74649	7	6	IR
197 and 211	2	5493	5493	33	0.797	TM
235	2	7284	7299	11	15	ID
219	2	13854	13886	9	22	I
201	2	14634	14647	14	2	I
232	2	16636	16850	12	16	I
216	2	17246	17259	12	5	I
240	2	17534	17546	13	25	I
212 and 179	2	22815	22815	21	0	TM
222	2	37139	37144	21	8	I
239	2	41469	41489	5	28	I
224	2	42090	42096	19	28	IR
234	2	44965	44973	18	15	I
203 and 95	2	52998	53005	13	0.797	TM
218	2	69588	69602	11	0.797	I
226	2	77317	77336	7	6	I

Table 13: The adjusted common tasks in $SS3^{*}_{C\text{-}40}$.

Task	Satellite	The earliest start time	The latest start time	Duration/s	Slewing angle/°	The way to adjust tasks
97	—	—	—	—	—	D
169	1	19825	19846	20	0	R
7	1	57459	57481	19	0	R
33	2	63000	63008	17	0	R

Table 14: The inserted emergency tasks in $SS3^{*}_{C\text{-}40}$.

Task	Satellite	The earliest start time	The latest start time	Duration/s	Slewing angle/°	The way to insert tasks
206	1	3240	3259	22	20	I
207 and 92	1	7327	7335	20	1.797	TM
208	1	9926	9950	17	15	I
238	1	11625	11661	6	30	I
223	1	12308	12342	7	15	I
150 and 210	1	12550	12570	21	0.797	TM
236	1	13088	13120	10	28	IR
231	1	14938	14966	13	20	I
205	1	22177	22208	11	23	ID
228	1	24349	24367	23	25	I
229	1	25154	25187	8	3	I
204	1	26570	26606	6	1	I
220 and 166	1	30146	30175	8	0.797	TM
213	1	34540	34570	13	9	I
214	1	57054	57083	12	2	I
221	1	58115	58141	16	29	I
227	1	58362	58382	21	6	I
209	1	58682	58709	15	10	I
202	1	60137	60164	15	3	I
217	1	63500	63529	12	29	I
230	1	64352	64382	12	4	I
215	1	74614	74649	7	6	IR
233	2	357	369	13	30	I
197 and 211	2	5493	5493	33	0.797	TM
235	2	7284	7299	11	15	I
219	2	13854	13886	9	22	I
201	2	14634	14647	14	2	I
232	2	16636	16850	12	16	I
216	2	17246	17259	12	5	I
240	2	17534	17546	13	25	I
212 and 179	2	22815	22815	21	0	TM
237	2	27709	27712	23	26	I
222	2	37139	37144	21	8	I
239	2	41469	41489	5	28	I
224	2	42090	42096	19	28	I
234	2	44965	44973	18	15	I
203 and 95	2	52998	53005	13	0.797	TM
218	2	69588	69602	11	3	I
226	2	77317	77336	7	6	I

Table 15: The adjusted common tasks in $SS3^{*}_{T\text{-}40}$.

Task	Satellite	The earliest start time	The latest start time	Duration/s	Slewing angle/°	The way to adjust tasks
97	—	—	—	—	—	D
135	—	—	—	—	—	D
169	1	19825	19846	20	0	R
7	1	57459	57481	19	0	R
33	2	63000	63008	17	0	R
106	2	69608	69610	22	0	R

Table 16: The inserted emergency tasks in $SS3^{*}_{T\text{-}40}$.

Task	Satellite	The earliest start time	The latest start time	Duration/s	Slewing angle/°	The way to insert tasks
206	1	3240	3259	22	20	I
235	1	8626	8657	11	30	I
208	1	9926	9950	17	15	I
238	1	11625	11661	6	30	I
223	1	12308	12342	7	15	ID
210 and 150	1	12550	12550	53	1.5	TM
236	1	13088	13120	10	28	IR
205	1	22177	22208	11	23	ID
228	1	24349	24367	23	25	I
229	1	25154	25187	8	3	I
204	1	26570	26606	6	1	I
220 and 166	1	30141	30141	47	1.5	TM
213	1	34540	34570	13	9	I
214	1	57054	57083	12	2	I
209	1	58682	58709	15	10	I
202	1	60137	60164	15	3	I
217	1	63500	63529	12	29	I
230	1	64352	64382	12	4	I
215	1	74614	74649	7	6	IR
233	2	357	369	13	30	I
211 and 197	2	5488	5488	54	1.5	TM
231	2	13241	13254	13	29	I
219	2	13854	13886	9	22	IR
232	2	16636	16850	12	16	I
216	2	17246	17259	12	5	I
240	2	17534	17546	13	25	I
212 and 179	2	22808	22808	32	1	TM
237	2	27709	27712	23	26	I
222	2	37139	37144	21	8	I
239	2	41469	41489	5	28	I
224	2	42090	42096	19	28	I
234	2	44965	44973	18	15	I
218	2	69588	69590	11	3	I
226	2	77317	77336	7	6	I

Table 17: The adjusted common tasks in $SS2^*_{C\text{-}60}$.

Task	Satellite	The earliest start time	The latest start time	Duration/s	Slewing angle/°	The way to adjust tasks
97	—	—	—	—	—	D
193	—	—	—	—	—	D
70	—	—	—	—	—	D
173	—	—	—	—	—	D
133	—	—	—	—	—	D
169	1	19825	19846	20	0	R
14	1	32626	32661	6	0	R
120	1	43300	43333	8	0	R
142	1	45221	45250	12	0	R
35	1	54447	54481	7	0	R
28	1	61374	61398	17	0	R
143	1	84955	84986	10	0	R
71	2	58332	58351	6	0	R
132	2	59727	59746	6	0	R
128	2	65466	65485	7	0	R
25	2	65838	65855	9	0	R

can be merged by the compact task merging method with lower energy cost.

When there are 40 emergency tasks, $SS3^*_{C\text{-}40}$ is found to have higher revenue and less disturbance compared with $SS2^*_{C\text{-}40}$. It indicates that the total task execution time is indeed a robust indicator that can reflect the ability of a scheduling scheme to accept new tasks. If the total task execution time is shorter, then there will be larger spare time which may provide more imaging opportunities for new tasks. In contrast with $SS3^*_{T\text{-}40}$, $SS3^*_{C\text{-}40}$ contains more emergency tasks that are inserted by compact task merging. This result proves that the strategy of compact task merging has advantage over the existing traditional method. The compact task merging method can improve the chance of merging multiple tasks.

When there are 60 emergency tasks, the superiority of RSSM3 appears to be more obvious. The disturbance of $SS3^*_{C\text{-}60}$ is much lower than that of $SS2^*_{C\text{-}60}$. This result supports the conclusion that the total task execution time is a useful robust indicator that can improve the robustness of schedules. Moreover, the compact task merging method can improve the chance of merging multiple tasks.

6. Conclusion

To address the dynamic scheduling problem of satellite observations, we constructed a two-stage solution for emergency response in the paper. The imaging duration is embedded in the two-stage solution in order to make full use of the satellite resources. In the first stage, the robust satellite scheduling model is established considering a new robust indicator of the total task execution time. The multiobjective genetic algorithm is used to solve the model. In the second stage, we proposed a new strategy to construct compact composite tasks using task execution duration. The compact task merging method is embedded in the rule-based heuristic algorithm, which is designed to adjust the initial schedules dynamically. To evaluate our model and algorithm, we performed experiments and compared the scheduling schemes generated by different methods. The results of experimental simulations validate the impact of the task execution duration. The comparisons and analysis performed in this study demonstrated that the duration of task execution is an important factor. When the total duration of task execution is an objective for optimization of the scheduling, it can improve the robustness of the produced schedules. Moreover, when the task execution duration is used in the compact task merging method, it can improve the observation opportunity of new tasks. Therefore, we conclude that the task execution duration has considerable significance for full utilizing of satellite resources.

Appendix

In the appendix, the experimental data of common tasks and emergency tasks are shown in Tables 3 and 4.

Tables 5–22 show the adjusted common tasks and inserted new tasks in schedules produced by different methods.

For the sake of simplifying the expressions, "TM," "I," "IR," and "ID" denote the four ways, that is, insertion by

Table 18: The inserted emergency tasks in $SS2^{*}_{C\text{-}60}$.

Task	Satellite	The earliest start time	The latest start time	Duration/s	Slewing angle/°	The way to insert tasks
254	1	697	721	17	7	I
206	1	3240	3259	22	20	IR
257	1	6341	6372	11	13	IR
207 and 92	1	7327	7335	20	1.797	TM
208	1	9926	9950	17	15	IR
210 and 150	1	12550	12570	21	0.797	TM
231	1	14938	14966	13	20	IR
247	1	16490	16523	10	28	I
227	1	18038	18058	21	8	ID
221	1	18271	18297	16	25	IR
246	1	21848	21884	5	3	I
205	1	22177	22208	11	23	ID
230	1	24092	24122	12	15	IR
228	1	24349	24367	23	25	I
229	1	25154	25187	8	3	I
46 and 204	1	26593	26606	23	0	TM
220	1	30146	30180	8	3	I
245	1	33626	33652	14	0	I
213	1	34540	34570	13	9	I
248	1	35849	35879	12	15	I
252 and 115	1	36145	36145	17	0	TM
255	1	55732	55764	10	4	I
214	1	57054	57083	12	2	I
209 and 258	1	58709	58709	26	7.797	IR,TM
202	1	60137	60164	15	3	I
217	1	63500	63529	12	29	I
250	1	72816	72847	10	8	I
215	1	74614	74649	7	6	IR
242	1	78044	78071	15	15	I
243	2	100	127	12	2	I
233	2	357	369	13	30	IR
249 and 96	2	725	730	21	0	I
260	2	2541	2547	19	4	I
211 and 197	2	5493	5493	33	0.797	TM
235	2	7284	7299	11	15	ID

Table 18: Continued.

Task	Satellite	The earliest start time	The latest start time	Duration/s	Slewing angle/°	The way to insert tasks
219	2	13854	13886	9	22	I
201	2	14634	14647	14	2	I
232	2	16636	16850	12	16	I
216	2	17246	17259	12	5	I
240	2	17534	17546	13	25	I
256	2	18779	18799	5	10	IR
212 and 179	2	22815	22815	21	0	TM
244	2	26087	26105	8	6	I
237	2	27709	27712	23	26	I
222	2	37139	37144	21	8	I
241	2	39695	39712	8	6	I
239	2	41469	41489	5	28	I
224	2	42090	42096	19	28	IR
234	2	44965	44973	18	15	I
251	2	47554	47565	14	12	I
238	2	48891	48911	6	25	ID
203 and 95	2	52998	53005	13	0.797	TM
223	2	53910	53928	7	6	I
218	2	69588	69602	11	0.797	I
259	2	76325	76333	18	3	I
226	2	77317	77336	7	6	I

Table 19: The adjusted common tasks in $SS3^{*}_{C\text{-}60}$.

Task	Satellite	The earliest start time	The latest start time	Duration/s	Slewing angle/°	The way to adjust tasks
97	—	—	—	—	—	D
173	—	—	—	—	—	D
7	—	—	—	—	—	D
169	1	19825	19846	20	0	R
133	1	58085	58099	22	0	R
33	2	63000	63008	17	0	R
128	2	65466	65485	7	0	R

Table 20: The inserted emergency tasks in $SS3^{*}_{C\text{-}60}$.

Task	Satellite	The earliest start time	The latest start time	Duration/s	Slewing angle/°	The way to insert tasks
254	1	697	721	17	7	I
206	1	3240	3259	22	20	I
257	1	6341	6372	11	13	I
92 and 203	1	7327	7335	20	1.797	TM
208	1	9926	9950	17	15	I
223	1	12308	12342	7	15	I
150 and 210	1	12550	12570	21	0.797	TM
236	1	13088	13120	10	28	ID
231	1	14938	14966	13	20	I
247	1	16490	16523	10	28	I
227	1	18038	18058	21	8	IR
246	1	21848	21884	5	3	I
205	1	22177	22208	11	23	ID
230	1	24092	24122	12	15	IR
228	1	24349	24367	23	25	I
229	1	25154	25187	8	3	I
242	1	28464	28491	15	242	I
220 and 166	1	30146	30175	8	0.797	TM
245	1	33626	33652	14	0	I
213	1	34540	34570	13	9	I
248	1	35849	35879	12	15	I
115 and 252	1	36145	36145	17	0	TM
255	1	55732	55764	10	4	I
214	1	57054	57083	12	2	I
221	1	58115	58141	16	29	I
209 and 258	1	58709	58709	26	7.797	TM
202	1	60137	60164	15	3	I
217	1	63500	63529	12	29	I
250	1	72816	72847	10	8	I
215	1	74614	74649	7	6	IR
243	2	100	127	12	2	I
233	2	357	369	13	30	I
96 and 249	2	725	730	21	0	TM
260	2	2541	2547	19	4	I
82 and 204	2	5237	5242	20	0	TM

Table 20: Continued.

Task	Satellite	The earliest start time	The latest start time	Duration/s	Slewing angle/°	The way to insert tasks
197 and 211	2	5493	5493	33	0.797	TM
235	2	7284	7299	11	15	I
219	2	13854	13886	9	22	IR
201	2	14634	14647	14	2	I
232	2	16636	16850	12	16	I
216	2	17246	17259	12	5	I
240	2	17534	17546	13	25	I
256	2	18779	18799	5	256	I
212 and 179	2	22815	22815	21	0	TM
244	2	26087	26105	8	6	I
237	2	27709	27712	23	26	I
222	2	37139	37144	21	8	I
241	2	39695	39712	8	6	I
239	2	41469	41489	5	28	I
224	2	42090	42096	19	28	I
234	2	44965	44973	18	15	I
251	2	47554	47565	14	12	I
238	2	48891	48911	6	25	ID
203 and 95	2	52998	53005	13	0.797	TM
218	2	69588	69602	11	3	I
259	2	76325	76333	18	3	I
226	2	77317	77336	7	6	I

Table 21: The adjusted common tasks in $SS3^{*}_{T\text{-}60}$.

Task	Satellite	The earliest start time	The latest start time	Duration/s	Slewing angle/°	The way to adjust tasks
97	—	—	—	—	—	D
173	—	—	—	—	—	D
40	—	—	—	—	—	D
236	—	—	—	—	—	D
169	1	19825	19846	20	0	R
133	1	58085	58099	22	0	R
33	2	63000	63008	17	0	R
128	2	65466	65485	7	0	R

Table 22: The inserted emergency tasks in $SS3^{*}_{T\text{-}60}$.

Task	Satellite	The earliest start time	The latest start time	Duration/s	Slewing angle/°	The way to insert tasks
254	1	697	721	17	7	I
206	1	3240	3259	22	20	I
257	1	6341	6372	11	13	I
208	1	9926	9950	17	15	I
150 and 210	1	12550	12550	53	1.5	TM
236	1	13088	13120	10	28	ID
231	1	14938	14966	13	20	ID
247	1	16490	16523	10	28	I
227	1	18038	18058	21	8	IR
246	1	21848	21884	5	3	I
205	1	22177	22208	11	23	ID
230	1	24092	24122	12	15	IR
228	1	24349	24367	23	25	I
229	1	25154	25187	8	3	I
242	1	28464	28491	15	242	I
220 and 166	1	30141	30141	47	1.5	TM
245	1	33626	33652	14	0	I
213	1	34540	34570	13	9	I
248	1	35849	35879	12	15	I
252	1	36153	36185	9	1	TM
255	1	55732	55764	10	4	I
214	1	57054	57083	12	2	I
221	1	58115	58141	16	29	I
209	1	58682	58709	15	10	TM
258	1	58721	58750	14	8	I
202	1	60137	60164	15	3	I
217	1	63500	63529	12	29	I
250	1	72816	72847	10	8	I
215	1	74614	74649	7	6	IR
259	1	75794	75820	18	4	I
226	1	76835	76870	7	25	I
243	2	100	127	12	2	I
233	2	357	369	13	30	I
249	2	766	776	10	0	TM
260	2	2541	2547	19	4	I
82 and 204	2	5237	5237	40	0.5	TM
235	2	7284	7299	11	15	I
219	2	13854	13886	9	22	IR
201	2	14634	14647	14	2	I
232	2	16636	16850	12	16	I
216	2	17246	17259	12	5	I
240	2	17534	17546	13	25	I
256	2	18779	18799	5	256	I
212 and 179	2	22808	22808	32	1	TM
244	2	26087	26105	8	6	I
237	2	27709	27712	23	26	I

Table 22: Continued.

Task	Satellite	The earliest start time	The latest start time	Duration/s	Slewing angle/°	The way to insert tasks
222	2	37139	37144	21	8	I
241	2	39695	39712	8	6	I
239	2	41469	41489	5	28	I
224	2	42090	42096	19	28	I
234	2	44965	44973	18	15	I
251	2	47554	47565	14	12	I
238	2	48891	48911	6	25	ID
223	2	53910	53928	7	6	I
218	2	69588	69602	11	3	I

task merging, direct insertion, insertion by rearranging, and insertion by deleting to insert new tasks. Besides, the adjusted common tasks are either rearranged "R" or deleted "D". The time (the earliest start time/the latest start time) is represented by the time span (in seconds) between real time and 7 March 2014 12:00:00.000.

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 Program for New Century Excellent Talents in University (no. NECT-11-0039), and the National Key Technology Research and Development Program of the Ministry of Science and Technology of China (no. 2012BAH27B01 and no. 2012BAH27B03).

References

[1] 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.

[2] G. Verfaillie, M. Lemaître, and T. Schiex, "Russian doll search for solving constraint optimization problems," in *AAAI/IAAI: Proceedings of the Thirteenth National Conference on Artificial Intelligence and the Eighth Annual Conference on Innovative Applications of Artificial Intelligence*, vol. 1, pp. 181–187, 1996.

[3] I. M. Ovacik and R. Uzsoy, "Rolling horizon algorithms for a single-machine dynamic scheduling problem with sequence-dependent setup times," *International Journal of Production Research*, vol. 32, no. 6, pp. 1243–1263, 1994.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] A. Globus, J. Crawford, J. Lohn, and A. Pryor, "Scheduling earth observing satellites with evolutionary algorithms," in *Proceedings of the Conference on Space Mission Challenges for Information Technology (SMC-IT '03)*, 2003.

[10] 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 Genetic and Evolutionary Computation Conference (GECCO '07)*, pp. 2211–2218, July 2007.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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, August 1994.

[16] 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.

[17] D. Qiu, C. He, J. Liu, and M. Ma, "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.

[18] B. Sun, W. Wang, and Q. Qi, "Satellites scheduling algorithm based on dynamic constraint satisfaction problem," in *Proceedings of the International Conference on Computer Science and Software Engineering (CSSE '08)*, pp. 167–170, December 2008.

[19] 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.

[20] 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 oftware Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing (SNPD '07)*, pp. 9–14, 2007.

[21] 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.

[22] Y. Xu, P. Xu, H. Wang, and Y. Peng, "Clustering of imaging reconnaissance tasks based on clique partition," *Operations Research and Management Science*, vol. 19, no. 4, pp. 143–149, 2010.

[23] 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.

[24] B.-C. Bai, Y.-W. Chen, R.-J. He, and J.-F. Li, "Scheduling satellites observation and task merging based on decomposition optimization algorithm," *Acta Automatica Sinica*, vol. 35, no. 5, pp. 596–604, 2009.

[25] 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.

[26] J. Wang, *Research on robust scheduling approach and its application of imaging satellites [Ph.D. thesis]*, College of Information System and Management, National University of Defense Technology, Changsha, China, 2008.

[27] 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.

[28] A. Konak, D. W. Coit, and A. E. Smith, "Multi-objective optimization using genetic algorithms: a tutorial," *Reliability Engineering & System Safety*, vol. 91, no. 9, pp. 992–1007, 2006.

[29] R. Deng, H. Tang, Y. Shan et al., "Emergency task oriented satellite robust mission planning model and algorithm," *Remote Sensing Information*, vol. 29, pp. 25–31, 2014.

[30] J. Frank, A. Jonsson, R. Morris, D. E. Smith, and P. Norvig, "Planning and scheduling for fleets of earth observing satellites," in *Proceedings of the 6th International Symposium on Artificial Intelligence, Robotics, Automation & Space*, June 2001.