Research Article

Heuristic Scheduling Algorithm Oriented Dynamic Tasks for Imaging Satellites

Maocai Wang,[1,2] Guangming Dai,[1] and Massimiliano Vasile[2]

[1] School of Computer, China University of Geosciences, Wuhan 430074, China
[2] Department of Mechanical and Aerospace Engineering, University of Strathclyde, Glasgow G1 1XJ, UK

Correspondence should be addressed to Maocai Wang; cugwmc@gmail.com

Imaging satellite scheduling is an NP-hard problem with many complex constraints. This paper researches the scheduling problem for dynamic tasks oriented to some emergency cases. After the dynamic properties of satellite scheduling were analyzed, the optimization model is proposed in this paper. Based on the model, two heuristic algorithms are proposed to solve the problem. The first heuristic algorithm arranges new tasks by inserting or deleting them, then inserting them repeatedly according to the priority from low to high, which is named IDI algorithm. The second one called ISDR adopts four steps: insert directly, insert by shifting, insert by deleting, and reinsert the tasks deleted. Moreover, two heuristic factors, congestion degree of a time window and the overlapping degree of a task, are employed to improve the algorithm's performance. Finally, a case is given to test the algorithms. The results show that the IDI algorithm is better than ISDR from the running time point of view while ISDR algorithm with heuristic factors is more effective with regard to algorithm performance. Moreover, the results also show that our method has good performance for the larger size of the dynamic tasks in comparison with the other two methods.

1. Introduction

Because earth observation satellites (EOS) have many features such as wide coverage area and long duration and are without boundaries limitation, they have become an important means for exploring and researching earth resources and have been widely used in the fields such as land surveying, vegetation classification, crop growth trend assessment, natural disaster monitoring, and management of large-scale infrastructure projects as well as battlefield reconnaissance and ground military target identification.

Mission planning plays a key role in the whole process of earth observation. It directly affects the result of task completion. With the increase of the types of on-orbit satellites, as well as the increasingly complex requirements for observation data, how to optimize the scheduling of satellite resources to meet all types of observational requests has presented new challenges for satellite mission planning.

Satellite scheduling is to allocate the observation resources and executing time to a series of imaging tasks.

In the recent years, many researchers have focused on different types of scheduling problems for EOS. For example, Parish [1] adopted the genetic algorithm to schedule as many supports as possible by a schedule builder program for 24-hour satellite range schedules. Wolfe and Sorensen [2] described the priority dispatch algorithm and the look ahead algorithm and then presented a novel genetic algorithm with two additional binary variables. Vasquez and Hao [3] formalized the daily photograph scheduling problem of EOS as a generalized version of the well-known knapsack model and developed a tabu search algorithm to solve the problem. Vasquez and Hao [4] also designed a partition-based approach to get the tight upper bounds for the daily photograph scheduling problem of EOS, and then a simplex-based linear programming relaxation and a relaxed knapsack approach were presented to solve the problem. In addition, Barbulescu et al. [5] compared simple heuristic method, local search method, and genetic algorithm and showed that the genetic algorithm had the best performance in the three algorithms for the larger and more difficult

problems. Frank et al. [6] used a constraint-based language to model the scheduling problem of EOS and put forward a stochastic greedy algorithm with heuristics search. Globus et al. [7, 8] compared some algorithms such as the genetic algorithm, hill climbing, simulated annealing, squeaky wheel optimization, and iterated sampling on two variants of a realistically-sized model of the EOS scheduling problem. Lin et al. [9] adopted Lagrangian relaxation and linear search techniques to generate a near-optimal, feasible schedule for satellite daily imaging scheduling problem. Also, Zufferey et al. [10] designed a tabu search and an adaptive memory algorithm to tackle the satellite range scheduling problem by employing the best ingredients of the graph coloring techniques to the problem. Wang [11] designed a dynamic insert tasks heuristic algorithm named as DITHA for imaging satellite scheduling problem, which comprised three basic processes: insert directly, insert by iteration method, and insert by replacing. Mansour and Dessouky [12] proposed a genetic algorithm to solve the SPOT5 selection problem by a new genome representation. Huang et al. [13] proposed a fast heuristic algorithm called FHTIA for electronic reconnaissance satellite dynamic scheduling problem, which comprised three basic operators: directly insert tasks algorithm, shift and insert tasks algorithm, and replace tasks algorithm. As well, Zhang et al. [14] presented an ant colony optimization approach with a guidance solution to avoid trapping in local optima for solving the satellite control resource scheduling problem. Wang et al. [15] presented a nonlinear model of the scheduling problem and developed a priority-based heuristic with conflict avoided to schedule EOS constellation. Marinelli et al. [16] developed a Lagrangian version of the Fix-and-Relax MIP heuristic to solve the large scale input variables for satellite scheduling problem. Sarkheyli et al. [17] modeled the scheduling problem as the graph coloring and proposed a new tabu search algorithm to solve resources scheduling in low earth orbit by a new move function. Wu et al. [18] presented a novel two-phase based scheduling method in task clustering phase and task scheduling phase, constructed an acyclic directed graph model, and utilized a hybrid ant colony optimization algorithm for satellite observation scheduling. Dishan et al. [19] constructed an integer programming model, designed the rolling horizon strategy, and proposed three dynamic scheduling algorithms with the rolling horizon strategy for the dynamic scheduling problem of EOS. Wu et al. [20] designed an acyclic directed graph model by constructing a subacyclic directed graph in each orbit for multisatellite scheduling problem oriented to emergency tasks and common tasks and then presented a hybrid ant colony optimization algorithm mixed with iteration local search by repairing the constraints to guarantee all solutions are satisfying the emergency task requirement. Wang et al. [21] proposed a rule-based heuristic algorithm by describing the dynamic factors such as the changes of task amount, task properties, and task constraints into a uniformed form as inserting new tasks for dynamic scheduling problem of earth observing satellites and designed two heuristic rules named as max-contention for retraction and min-occupation for insertion, respectively. Iacopino et al. [22] designed an

innovative dynamic planning algorithm with a high-level of adaptability and responsiveness, and the algorithm was based on ant colony algorithm by applying stigmergy to find near-optimum solutions on a global level for the Disaster Monitoring Constellation.

This paper is organized as follows. After analyzing the dynamic properties of imaging satellite scheduling, the optimization model is proposed. Then, two heuristic algorithms and two heuristic factors are put forward to solve the problem. A description of the test example follows. After presenting the results of the tests, the ability and the applicability of the heuristic algorithms and the heuristic factors are analyzed to derive some conclusions.

2. Analysis for the Dynamic Properties

Imaging satellites scheduling problem may face many dynamic factors as follows.

(i) Insert a new task: according to the actual requirement, users may insert some new tasks when a scheduling scheme is being executed, especially some unexpected tasks, such as the monitoring of serious earthquakes and volcanic eruptions and forest fires.

(ii) Cancel some tasks arranged: due to the change of user requirements, some arranged tasks may be cancelled before they are scheduled.

(iii) Change the task properties: some tasks may change their properties because users change their requirements or because task attributes are not reasonable.

(iv) Uncertainty of weather conditions: some tasks arranged may not be completed or have poor quality due to the change of the weather conditions such as cloud cover.

In fact, except for the dynamic properties caused by the tasks, there are many dynamic uncertainties resulting from the resources. For example, some satellites may be out of use because of malfunction, or some satellites may change their attitudes or orbit parameters, which results in the dynamic changes from the resources. However, the change of the resources can be seen as a change of the task. If some tasks are arranged to be executed in advance by a satellite and the satellite is invalid later, these tasks can be seen as new insert tasks. Therefore, we research the dynamic scheduling only in the change of the tasks.

3. Model

This study mainly focuses on imaging satellite scheduling problem of point targets, which means each target is viewed as a point that can be observed by a single observation strip. In fact, the scheduling problem of area targets can be solved by dividing them into many point targets.

Different to the static scheduling, in addition to the tradition requirements, there are some new requirements for the dynamic scheduling. The common requirements for the dynamic scheduling are as follows.

As far as possible to maximize the observation income: the optimal observation income is the initial goal of scheduling scheme and is also the goal always pursued in the whole scheduling process. This goal is often difficult to achieve because of a variety of reasons. The complexity of the problem and the limitation of computing ability result in the difficulty to achieve the optimal scheduling income. So the decision makers often have to accept a satisfied solution, not the optimal solution.

As far as possible to complete all dynamic tasks: in general, the dynamic tasks are the emergency tasks, which need to be completed in time. Therefore, in dynamic scheduling, the first and foremost thing is to complete the dynamic tasks immediately when they are added to the task set.

As far as possible to minimize the difference between the new scheme and the old one: satellite application is a complicated process, and it needs special equipment and takes time to upload an instruction to the satellite. Once the scheduling scheme is adjusted on a large scale, there may be a series of influences on the decision. Although the better income may be gotten by rescheduling fully, it may lead to a large-scale adjustment for the scheduling scheme, which results in the phenomenon of "shock" for the scheduling scheme.

As far as possible to speed up the adjustment from the old scheme to the new one: in the actual process of imaging satellites scheduling, the requirement for the timeliness of dynamic adjustment is very high. If it takes too long time to adjust the scheme, maybe the new scheme generated just does not meet the requirement. The goal with the optimal income usually conflicts with the one with the fastest adjustment. Although the optimal income can be gained by rescheduling, the new scheme is usually difficult to meet the timeliness requirements.

3.1. Parameters. The scheduling problem in the paper is characterized by the following sets, input parameters, and decision variables.

Sets:

S: the set of satellites

T: the set of tasks

DT: the set of dynamic tasks

W: the set of the time windows

W^j: the set of the time windows observed by satellites j

W_i^j: the set of the time windows observed by satellite j for task i.

Parameters:

m: the number of the satellites

n: the number of all tasks

sn: the number of the tasks scheduled

dn: the number of the dynamic tasks

swn: the number of the time windows scheduled for static tasks

c_i: the observed income of task i

n_i^j: the number of the visible time windows of task i observed by satellite j

$w_{i,k}^j$: the kth time windows of task i observed by satellite j

$st_{i,k}^j$: the start time the kth time windows observed by satellite j for task i

$et_{i,k}^j$: the end time the kth time window observed by satellite j for task i

$ang_{i,k}^j$: the swing angle when task i is observed by satellite j in its kth time window

pre_j: the setup time of satellite j for starting

$shutd_j$: the stable time of satellite j for shutdown

lim_j: the maximum allowable running time of satellite j for each loop

$t\,min_j$: the shortest running time for satellite j

$t\,max_j$: the longest running time for satellite j.

Decision variables:

$x_{i,k}^j$: a Boolean variable that can be either 1 if the task i can be observed by the satellite j in the kth time window or 0 if not observed.

It is noted that there may be many visible time windows for every task observed by each satellite. We denote the number of the visible time windows of task i observed by satellite j as $|W_i^j|$.

3.2. Optimal Model. The optimization model proposed in the paper consisted of two optimization objectives. The first one is to complete dynamic tasks as many as possible. This objective is particularly important for the emergency scheduling. The second one is to maximize the observation income, which is the common objective for all imaging scheduling. In the model, three constraints are considered. They are as follows:

$$\text{Max. } \sum_{i=1}^{dn}\sum_{j=1}^{m}\sum_{k=1}^{|W_i^j|} x_{i,k}^j,$$

$$\text{Max. } \sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=1}^{|W_i^j|} x_{i,k}^j c_i \tag{1}$$

St. $x_{i,k}^j t\min_j \leq x_{i,k}^j \left(et_{i,k}^j - st_{i,k}^j\right) \leq x_{i,k}^j t\max_j,$

$$\forall i \in T, \quad j \in S, \quad k \in W_i^j \tag{2}$$

$\left[st_{i,k}^j - et_{i',k'}^j - \left(pre_j + shutd_j + \right.\right.$

$\left.\left. Tran\left(ang_{i,k}^j, ang_{i',k'}^j\right)\right)\right] x_{i,k}^j x_{i',k'}^j \geq 0,$

$$\forall i, i' \in T, \quad s \in S, \quad k \in W_i^j, \tag{3}$$

$$k' \in W_{i'}^j, \quad st_{i,k}^j > et_{i',k'}^j$$

$$\sum_{j=1}^{|S|} \sum_{k=1}^{|W_i^j|} x_{i,k}^j \leq 1, \quad \forall i \in T, \tag{4}$$

where $Tran(ang_{i,k}^j, ang_{i',k'}^j)$ means the switching time that the same satellite j adjusts its swing angle from $ang_{i,k}^j$ to $ang_{i',k'}^j$ for completing task i' after having completed task i when task i and task i' are adjacent.

Constraint (2) means that each available time window must be longer than the shortest running time of its observing satellite and shorter than the longest running time of the satellite; constraint (3) denotes that any two observation time windows must satisfy the switch time requirement; constraint (4) shows that each task is observed no more than one time by all satellites.

4. Scheduling Strategy

The satellite scheduling is a complex constraint optimal problem with many constraints. Moreover, the problem has been turned out to be an NP-hard problem. In fact, Barbulescu et al. [23] showed that the single resource range scheduling problem is equivalent to the problem of minimizing the number of tardy jobs on a single-machine with release dates, which enables the deterministic algorithms not suitable to solve the problem if the input size of the problem becomes larger. In fact, there are hundreds of input variables only in one day observation for the satellite scheduling problem. Therefore, in almost all satellite scheduling, nondeterministic algorithms are adopted to solve the problem.

In order to meet the high timeliness requirement of dynamic satellite scheduling, two heuristic algorithms and two heuristic factors are proposed in the paper. In the two heuristic algorithms, calculating the time windows between the satellites and the targets is a basic procedure. In our method, the calculating process includes three steps. Firstly, the geocentric angle is computed according to the position of the satellite and the location of the target. Then the direction of the swing angle towards left or right is determined based on the relation between the location of the target and the normal vector of the satellite orbital plane. Finally, the exact visible time windows between a given satellite and a certain target are computed by iterating based on dichotomy. The detailed process to calculate the time windows had been given in our previous paper [24].

4.1. IDI Algorithm. In the first heuristic algorithm proposed, new dynamic tasks are inserted directly or inserted by deleting repeatedly, which is named as IDI. All new tasks are firstly inserted to a waiting queue according to their priorities from low to high. Note that the new tasks are sorted according to their priorities from low to high, not from high to low. What is the purpose? It is to ensure that the low priority new tasks can have the chance to be arranged while the high priority tasks can be arranged by eliminating the low priority tasks. If the high priority tasks are firstly arranged, the low priority tasks will not have the chance to be arranged because the high priority tasks occupied the only time windows of the low priority tasks. Therefore, if the priorities are sorted from low to high, the low priority tasks are firstly arranged and then, when the high priority tasks are considered, they will be arranged if there is a free time window; even without a free time window, they can also be arranged by deleting the low priority tasks (Algorithm 1).

Then the first task in the queue will be inserted to a free time window directly if it does not conflict with any other task. If inserting directly fails, some tasks will be deleted for inserting the first task in the queue. The choice for the deleted task will depend on an evaluation function. The evaluation function is based on the conflict degree. Every task has a conflict degree δ, which is defined as the number of the tasks that conflict with the task. The evaluation function based on the conflict degree is defined as follows:

$$f_i = \frac{c_i}{\left(1 + \delta_i\right)^2}, \tag{5}$$

where c_i denotes the observed income of task I and δ_i the conflict degree of task i. In the formula, the denominator $(1 + \delta_i)^2$ is designed to avoid the conflict by enlarging the punishment.

The deleted task will be placed in the rear of the queue. Then all the process will repeat until the waiting queue is empty.

4.2. ISDR Algorithm. In the second heuristic algorithm proposed, a new task is scheduled by three ways: insert directly, insert by shifting, and insert by deleting, and then the deleted tasks are rearranged; here we name the algorithm as ISDR. If a new task does not conflict with other tasks, the task can be inserted directly. If inserting a new task directly fails, which means the new task conflicts with other tasks, the task can be inserted by shifting other tasks. If inserting directly and inserting by shifting do not succeed, some tasks will be deleted for inserting a new task. The deleted tasks will be added to a waiting queue. After all new tasks have been inserted, the deleted tasks in the waiting queue will be sorted according to their priority. Then the tasks will be scheduled in turn if there are intervals. By ISDR, the new tasks, especially emergency tasks, can quickly be inserted to the initial scheduling scheme (Algorithm 2).

Figure 1 is an example about inserting by shifting. As is shown in Figure 1, every task has two time windows. One is the visible time windows and the other one is the available time windows. As for task i, $st_{i,k}^j$ and $et_{i,k}^j$ are the start time and the end time of a visible time window observed by the
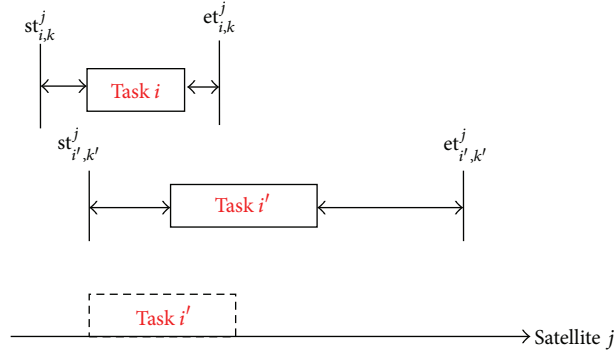
FIGURE 1: Insert by shifting.

(1) Set values for m, dn
(2) Sort dynamic task set DT according to the priorities from low to high
(3) for all $i \in [1, \ldots, \text{dn}]$ do
(4)   for all $j \in [1, \ldots m]$ do
(5)   Compute all visible time windows of task i observed by satellite j
(6)   Compute the number n_i^j of the visible time windows
          of task i observed by satellite j
(7)   end for
(8) end for
(9) while DT is not empty do
(10)   Take the serial number i of the first task in the queue
(11)   for all $j \in [1, \ldots, m]$ do
(12)   for all $k \in [1, \ldots, n_i^j]$ do
(13)   if the time windows$[\text{st}_{i,k}^j, \text{et}_{i,k}^j]$ does not conflict with others then
(14)   Inserting the time windows$[\text{st}_{i,k}^j, \text{et}_{i,k}^j]$ directly
(15)   goto Step 9.
(16)   end if
(17)   end for
(18) end for
(19) for all $j \in [1, \ldots, m]$ do
(20)   for all $k \in [1, \ldots, n_i^j]$ do
(21)   Caluculate the conflict degree of $w_{i,k}^j$ with all time
        windows of the tasks scheduled
(22)   end for
(23) end for
(24) Selecting the minimum conflict degree, take the serial
     number of the satellite as min j and the serial number of the time window as min k
(25) Deleting these scheduled tasks which time window
     conflict with the time window $w_{i,\min k}^{\min j}$
(26) Adding them to the rear of the queue
(27) Inserting the time window $w_{i,\min k}^{\min j}$ to the scheduled queue for task i
(28) end while

ALGORITHM 1: The algorithm of IDI.

satellite j. Similarly, $\text{st}_{i',k'}^j$ and $\text{et}_{i',k'}^j$ are the the start time and the end time of a visible time window of task i' observed by the same satellite j. The rectangle denotes the available time windows. Only if the actual time window of one task is arranged between the start time and the end time of its visible time windows and there is no conflict with other tasks, the actual time windows are the available time windows. As for task i', if the actual start time is located between the start

time $\text{st}_{i',k'}^j$ of the visible time windows and the end time of the actual time windows of task i, there is a conflict between task i and task i' because the time windows overlap. So the conflict can be eliminated by shifting the actual start time of task i'.

4.3. Heuristic Factors. In multisatellite scheduling problem, each task may have many available time windows. Therefore, when there are a few available time windows for some task,

(1) Set values for m, dn
(2) Set wn = 0
(3) Sort dynamic task set DT according to the priorities from high to low
(4) for all $i \in [1, \ldots, \text{dn}]$ do
(5)   for all $j \in [1, \ldots, m]$ do
(6)   Compute all visible time windows of task i observed by satellite j
(7)   Compute the number n_i^j of the visible time windows of task i observed by satellite j
(8)   end for
(9) end for
(10) for all $i \in [1, \ldots, \text{dn}]$ do
(11)   for all $j \in [1, \ldots, m]$ do
(12)   for all $k \in \left[1, \ldots, n_i^j\right]$ do
(13)   if $w_{i,k}^j$ not conflict with other time windows then
(14)   Inserting $w_{i,k}^j$ directly
(15)   goto Step 10.
(16)   end if
(17)   end for
(18)   end for
(19) for all $j \in [1, \ldots, m]$ do
(20)   for all $k \in \left[1, \ldots, n_i^j\right]$ do
(21)   for all $t \in [1, \ldots, \text{sn}]$ do
               if $w_{i,k}^j$ can inserted by shifting the time window of the tth task scheduled
(22)   Insert $w_{i,k}^j$ by shifting the time window of the tth task scheduled
(23)   goto Step 10.
(24)   end if
(25)   end for
(26)   end for
(27) end for
(28) for all $j \in [1, \ldots, m]$ do
(29)   for all $k \in [1, \ldots, n_i^j]$ do
(30)   for all $t \in [1, \ldots, \text{sn}]$ do
(31)   if the priority of the task t lower than the priority of task i then
(32)   if $w_{i,k}^j$ can be inserted by deleting the time windows of task t then
(33)   Insert $w_{i,k}^j$ by deleting the time windows of task t
(34)   Add the task t to the waiting task list
(35)   wn++
(36)   goto Step 10.
(37)   end if
(38)   end if
(39)   end for
(40)   end for
(41) end for
(42) end for
(43) for all $i \in [1, \ldots, \text{wn}]$ do
(44)   for all $j \in [1, \ldots, m]$ do
(45)   for all $k \in [1, \ldots, n_{i,k}^j]$ do
(46)   if $w_{i,k}^j$ not conflict with other tasks scheduled then
(47)   Inserting $w_{i,k}^j$ directly
(48)   goto *Step* 10
(49)   end if
(50)   end for
(51)   end for
(52) end for

ALGORITHM 2: The algorithm of ISDR.

how to select the appropriate time window is a very important problem. If some reasonable heuristic factors can be adopted to find the best time window for every task, the performance of the scheduling algorithm can be improved remarkably.

In this paper, we proposed two heuristic factors, one is to select new task based on the congestion degree between the time windows. The other one is to minimize the overlapping degree.

The congestion degree of a time window for some task is defined as the number of the tasks which could not be scheduled because they are affected by the task when it is scheduled in the time windows. When a task is scheduled, the congestion degree of every time window for the task is computed. Then all of the time windows for the task are sorted depending on the congestion degree in the order from small to large. For every task, the actual time window is selected according to the congestion degree.

Given the satellite, the overlapping degree of a task in a certain time point t is defined as the number of the tasks, which can also start in the time point. If task i will be observed by the satellite in the time windows $[\text{st}_{i,k}^{j}, \text{et}_{i,k}^{j}]$, the overlapping degree of task i on every time point will be computed. For any point, if the task i can be arranged in the time point, we denote $pck_i = 1$. The overlapping degree of task i on the time point t denotes as $OD_i = \sum_{j \in DT, j \neq i} pck_j$. Once the overlapping degree is computed for every time point, the actual time windows for a task can be selected according to the overlapping degree. The lower the overlapping degree, the more the chances selected.

4.4. Comparing with Other Methods. Wang et al. [21] designed two heuristic rules named as max-contention for retraction and min-occupation for insertion, respectively. The basic idea of max-contention is, for each time window, to compute the contention of every task that can be observed in the window and as far as possible to give priority to the task that only has the time windows, no other windows. The rule can guarantee to arrange the tasks that have few time windows. However, the rule does not consider whether the tasks with low priority have the chance to be arranged. Moreover, it does also not consider the conflicts of the time window among the tasks, which maybe result in many other tasks not being arranged because the task is observed. As for the rule of min-occupation, when inserted, a task is arranged in the place with minimal occupation for the time windows. By computing the occupation for every time window, the rule considers the occupation for the time window between the task and other tasks. However, when inserting, the rule is only based on the occupation, and the priority of the tasks is not considered, which results in some emergency tasks with high priority not having any chance to be arranged because they occupied a large period in the time window.

Based on the independent arriving time and deadline of the imaging, Dishan et al. [19] proposed three algorithms named as AIS/RH, DIS/RH, and WIS/RH by combining with rolling horizon. The algorithms consider the arrival sequence, the deadline sequence, and the waiting sequence for new tasks. The main idea is first arrangement for early arrival

or deadline tasks, which can ensure that the tasks with the earlier arrived time or deadline are observed. However, the algorithms do not consider the task priority and also do not involve the occupation or contention. Therefore, the algorithms cannot ensure the real time response for emergency tasks. Moreover, it is possible for some task to occupy a certain resource for a long time or result in many other tasks failure, which deteriorates the overall performance.

The algorithms proposed in this paper, both IDI algorithm and ISDR algorithm, are based on inserting directly or inserting by deleting. Moreover, two heuristic factors, the congestion degree or the overlapping degree, are designed to deal with the occupation and contention. As for the tasks to be inserted, they are sorted according to the priority from low to high, not from high to low, which ensures that the tasks with low priority have the chance to be observed. It depends on the conflict degree to decide which task will be deleted in order to insert a new task. The basic idea can ensure that the tasks deleted are the tasks that have most impacts on other tasks. Therefore, many tasks can be observed if the task is deleted, which improves the overall observation income.

5. Test

In the test, there are 145 static tasks selected from China cities. The initial scheduling scheme for the 145 static tasks is generated by genetic algorithm. Dynamic tasks are generated randomly in the longitude range from 75 to 120 and the altitude range from 20 to 45. In the test, five group dynamic tasks with different size are given. There are 11, 21, 61, 101, and 201 dynamic tasks in every group, respectively. In order to analyze the performance of the heuristic algorithms and the heuristic factors, seven group tests are designed depending on the different heuristic strategies, which are named as T1, T2, T3, T4, T5, T6, and T7. They denote the following means:

> T1: IDI algorithm without any heuristic factor
>
> T2: IDI algorithm with congestion degree
>
> T3: IDI algorithm with congestion degree and the overlapping degree
>
> T4: add the overlapping degree in the process inserting directly
>
> T5: add the overlapping degree in the process inserting directly and inserting by shifting
>
> T6: add the overlapping degree in the process inserting directly, inserting by shifting and by deleting
>
> T7: ISDR without any heuristic factor.

The test results are shown from Table 1 to Table 5.

As is shown in Table 1, if there are only a few dynamic tasks, whether there is any heuristic factor is not important; that means the heuristic factor have a little effect for the small size dynamic tasks. The main reason is to have little conflicts when the size of the dynamic tasks is small, and all new tasks can be scheduled. For the running time, it is clear that the tests with some heuristic factors or some strategy of ISDR will be slower than the test without any heuristic

TABLE 1: The result for 11 dynamic tasks with 145 static tasks.

Test item	Number of dynamic tasks completed	Number of static tasks completed	Number of static tasks affected	Income	Time (s)
T1	11	145	0	2726	0.093
T2	11	145	0	2726	0.094
T3	11	145	0	2726	0.5
T4	11	145	0	2726	1.093
T5	11	145	0	2726	0.436
T6	11	145	0	2726	0.421
T7	11	145	0	2726	0.031

TABLE 2: The result for 21 dynamic tasks with 145 static tasks.

Test item	Number of dynamic tasks completed	Number of static tasks completed	Number of static tasks affected	Income	Time (s)
T1	21	145	2	3641	0.438
T2	21	144	2	3631	0.719
T3	21	144	2	3631	1.875
T4	21	145	2	3641	2.969
T5	21	145	2	3641	1.139
T6	21	145	2	3641	1.139
T7	20	144	3	3544	3.869

strategy because there are many loops and judgments in ISDR algorithm, which consumes a lot of time. It is interesting to note that T1 is slower than T7; maybe there are too many evaluations to decide which task should be deleted in T1. It can also be found that T4 is slower than T5 and T6.

As is shown in Table 2, if one or many strategies in ISDR are adopted, the running time is longer. However, the number of the tasks completed in T4, T5, and T6 are more than the number in T2 and T3, which shows that the result of ISDR is better than IDI when the size of the dynamic tasks increases. It should be noted that T1 has better result than T2 and T3 although there is no any heuristic factor or heuristic strategy in T1, which shows the heuristic factors do not work because there are almost no conflicts when the size of the new tasks is small. Clearly, comparing T7 with T4, T5 and T6, ISDR without any heuristic factor is poorer than ISDR with heuristics, which shows that heuristic factors play an important role in ISDR. It is also shown there is no obvious difference in each process heuristic factors are added. The results in T4, T5, and T6 are closer to the global optimum than those in T2 and T3, which shows ISDR improves the effect of the global optimal results.

As is shown in Table 3, from the running time point of view, IDI is rapider than ISDR. However, the optimal result is better in ISDR than in IDI by comparing T4, T5, and T6 with T1, T2, and T3. With the increase of the number of the dynamic tasks, IDI with heuristic factor is better than those without any heuristic factor because there are more conflicts by comparing with T1, T2, and T3. It also can be noted that ISDR without any heuristic factor is significantly poorer than ISDR with heuristic factors, which shows heuristic factors play an important role in ISDR again. This conclusion is

consistent with the result in Table 2. Moreover, the heuristic factors added in the three processes of ISDR have little effect on the original scheduling scheme.

The results in Tables 4 and 5 also support the conclusion that IDI is better than ISDR from the running time point of view. Similarly, the other conclusions in Tables 1, 2, and 3 can be verified from Tables 4 and 5. That is to say, the optimal result is better in ISDR than in IDI. Moreover, with the increase of the number of the dynamic tasks, IDI with heuristic factor is better than those without any heuristic factor. Finally, ISDR without any heuristic factor is significantly poorer than ISDR with heuristic factors.

We also tested and compared our method with the methods proposed by Wang et al. [21] and Dishan et al. [19]. The test results are given in Tables 6 and 7.

In Table 6, there are 101 dynamic tasks. Obviously, the three methods have good performance for the test. It can also be seen that Qiu's method outperforms our method and Wang's method both in the performance and in the running time for the case with 101 dynamic tasks. It should be noted that all of 101 dynamic tasks and 145 static tasks are completed in Qiu's method, which shows that the method based on the idea of first arrangement for early arrival or deadline tasks is effective when the size of the new tasks is small because there are few conflicts between the time windows.

In Table 7, there are 201 dynamic tasks. With the increase of the number of the dynamic tasks, the performance of Qiu's method worsens rapidly, and many dynamic tasks could not be completed because of the conflicts. In the three methods, our method outperforms, and all dynamic tasks are completed in our method, which shows that our method is effective when the size of the new tasks is larger because

TABLE 3: The result for 61 dynamic tasks with 145 static tasks.

Test item	Number of dynamic tasks completed	Number of static tasks completed	Number of static tasks affected	Income	Time (s)
T1	61	141	9	6969	3.172
T2	61	142	7	6979	2.391
T3	61	142	7	6979	7.218
T4	61	145	8	7010	14.187
T5	61	145	6	7010	13.572
T6	61	145	6	7010	5.648
T7	58	138	17	6716	14.243

TABLE 4: The result for 101 dynamic tasks with 145 static tasks.

Test item	Number of dynamic tasks completed	Number of static tasks completed	Number of static tasks affected	Income	Time (s)
T1	101	134	17	10299	7.562
T2	101	136	16	10321	7.313
T3	101	137	15	10332	18.766
T4	101	144	17	10406	110.187
T5	101	142	22	10386	150.976
T6	101	143	32	10396	97.069
T7	93	125	33	9600	80.184

TABLE 5: The result for 201 dynamic tasks with 145 static tasks.

Test item	Number of dynamic tasks completed	Number of static tasks completed	Number of static tasks affected	Income	Time (s)
T1	201	105	53	18441	38.422
T2	201	112	51	18518	37.032
T3	201	109	50	18468	89.875
T4	198	123	50	18332	75.578
T5	201	120	57	18621	210.1
T6	201	131	67	18740	315.383
T7	168	103	57	15867	113.178

TABLE 6: Comparison for three methods (101 dynamic tasks).

Test item	Number of dynamic tasks completed	Number of static tasks completed	Number of static tasks affected	Income	Time (s)
We	101	143	32	10396	97.069
Wang	101	117	57	9833	107.596
Qiu	101	145	11	10402	77.391

TABLE 7: Comparison for three methods (201 dynamic tasks).

Test item	Number of dynamic tasks completed	Number of static tasks completed	Number of static tasks affected	Income	Time (s)
We	201	131	67	18740	315.383
Wang	200	107	84	13692	437.169
Qiu	163	139	41	15771	201.447

ISDR with the overlapping degree inserts new tasks as much as possible by inserting directly and shifting or deleting then inserting. It can be noted that Wang's method is also effective for the case of 201 dynamic tasks, which shows that the rules of the max-contention for retraction and the min-occupation for insertion play an important role.

As for the running time, as is shown in Tables 6 and 7, Qiu's method is fastest in the three methods, maybe because the rule of first arrangement for early arrival or deadline tasks is simple. Our method and Wang's method are slower because there are many judgments in the two methods.

6. Conclusion

In this paper, two heuristic algorithms are designed for the dynamic scheduling of imaging satellite. In one algorithm, new dynamic tasks are inserted directly or inserted by deleting firstly while there are four steps for scheduling the new dynamic tasks in another algorithm: insert directly, insert by shifting, insert by deleting, and reinsert for the tasks deleted. Moreover, two heuristic factors are also adopted to improve the algorithm's performance. The result shows IDI is better than ISDR from the running time point of view, and IDI is especially suitable for solving the problem with large-size dynamic tasks while the high timeliness is the principal consideration. On the contrary, ISDR with heuristic factors outperforms IDI with regard to algorithm performance, but it takes longer time.

In future work, we should be concerned with more real world problems by taking into account more complex constraints such as weather condition and lighting condition. In addition, ISDR algorithm needs to be improved for speeding up its running time.

Conflict of Interests

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

Acknowledgments

References

[1] D. A. Parish, *A genetic algorithm approach to automating satellite range scheduling [Ph.D. thesis]*, Air Force Institute of Technology Air University, 1994.

[2] 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.

[3] 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. An International Journal*, vol. 20, no. 2, pp. 137–157, 2001.

[4] 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.

[5] L. Barbulescu, A. Howe, J. P. Watson, and D. Whitley, "Satellite range scheduling: a comparison of genetic," in *Proceedings of the 7th International Conference on Parallel Problem Solving from Nature*, 2002.

[6] J. Frank, A. Jansson, R. Morris, and D. Smith, "Planning and scheduling for fleets of earth observing satellites," in *Proceedings of the 6th International Symposium on Artificial Intelligence, Robotics, Automation and Space*, pp. 18–22, Montreal, Canada, 2002.

[7] A. Globus, J. Crawford, J. Lohn, and A. Pryor, "A comparison of techniques for scheduling earth observing satellites," in *Proceedings of the 19th National Conference on Artificial Intelligence*, pp. 836–843, San Jose, Calif, USA, July 2004.

[8] A. Globus, J. Crawford, J. Lohn, and A. Pryor, "Scheduling earth observing satellite with evolutionary algorithms," in *Proceedings of International Conference on Space Mission Challenges for Information Technology*, Pasadena, Calif, USA, 2003.

[9] W. Lin, D. Liao, C. Liu, and Y. Lee, "Daily imaging scheduling of an earth observation satellite," *IEEE Transactions on Systems, Man, and Cybernetics A: Systems and Humans*, vol. 35, no. 2, pp. 213–223, 2005.

[10] N. Zufferey, P. Amstutz, and P. Giaccari, "Graph colouring approaches for a satellite range scheduling problem," *Journal of Scheduling*, vol. 11, no. 4, pp. 263–277, 2008.

[11] J. Wang, *Research on robust scheduling approach and its application of imaging satellite [thesis]*, China National University of Defense Technology, 2008.

[12] M. A. A. Mansour and M. M. Dessouky, "A genetic algorithm approach for solving the daily photograph selection problem of the SPOT5 satellite," *Computers and Industrial Engineering*, vol. 58, no. 3, pp. 509–520, 2010.

[13] X. Huang, L. Zhang, D. Qiu, and J. Zhu, "Fast heuristic algorithm for electronic reconnaissance satellite dynamic scheduling problem," *Journal of Chinese Computer Systems*, pp. 31–11, 2010.

[14] N. Zhang, Z. Feng, and L. Ke, "Guidance-solution based ant colony optimization for satellite control resource scheduling problem," *Applied Intelligence*, vol. 35, no. 3, pp. 436–444, 2011.

[15] 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 and Industrial Engineering*, vol. 61, no. 2, pp. 322–335, 2011.

[16] F. Marinelli, S. Nocella, F. Rossi, and S. Smriglio, "A Lagrangian heuristic for satellite range scheduling with resource constraints," *Computers and Operations Research*, vol. 38, no. 11, pp. 1572–1583, 2011.

[17] A. Sarkheyli, A. Bagheri, B. Ghorbani-Vaghei, and R. Askari-Moghadam, "Using an effective tabu search in interactive resources scheduling problem for LEO satellites missions," *Aerospace Science and Technology*, vol. 29, no. 1, pp. 287–295, 2013.

[18] 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.

[19] 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.

[20] 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.

[21] 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, Qingda, China, August 2007.

[22] C. Iacopino, P. Palmer, N. Policella, A. Donati, and A. Brewer, "Highly responsive MPS for dynamic EO scenarios," in *Proceedings of the 12th International Conference on Space Operations (SpaceOps '12)*, Stockholm, Sweden, June 2012.

[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] M. Wang, G. Chen, G. Dai, and Z. Song, "Research on scheduling algorithm for multi-satellite and point target task on swinging mode based on evolution algorithm," *Journal of Computer Applications*, vol. 33, no. 11, pp. 3144–3148, 2013.