Emergency Scheduling of Multiple Imaging Satellites with Dynamic Merging

Jianjiang Wang[1] and Xiaomin Zhu[2]
National University of Defense Technology, Changsha, China, 410073

Dishan Qiu[3]
National University of Defense Technology, Changsha, China, 410073

Scheduling plays a significant role in improving observation effectiveness of imaging satellites. Although extensive satellite scheduling algorithms have been proposed, none of them focuses on emergency scheduling in dynamic environment. In this paper, a novel dynamic emergency scheduling model of multiple imaging satellites is established. To improve user's satisfaction ratio and resource utilization, we propose a novel task dynamic merging strategy, developing an alternative task set establishment (ATSE) algorithm used for dynamic merging. Besides, a novel dynamic emergency scheduling algorithm called DM-DES is presented, which considers the dynamic merging. To demonstrate the superiority of our DM-DES, we conduct extensive experiments by simulations to compare DM-DES with an existing algorithm—RBHA as well as a baseline algorithm—DES. The experimental results indicate that DM-DES improves the scheduling quality of others and is suitable for emergency scheduling.

I. Introduction

IMAGING satellites are the platforms equipped with optical sensors that orbit the earth to take photographs of special areas at the request of users.[1] Recently, taking photographs in emergency by imaging satellites has become a critical measure to get the first-hand information.[2] For example, when an earthquake occurs, the images of stricken areas can be obtained by imaging satellites. Importantly, the images are expected to be acquired within a few hours or even in dozens of minutes for timely conducting damage assessment and planning rescue policies. From this scenario, it is easy to conclude that emergency tasks are with short users' expected finish times but not strict restrictions, rather than the deadlines of real-time tasks, i.e., even though the finish time of a task is a bit later than its user's expectation, the task execution is still valuable. As the imaging of stricken areas, missing user's expected finish time may result in inferior rescue efficiency, but it is still worthy getting the satellite images.

On the other hand, there exist some uncertainties while taking photographs by imaging satellites including users' requirements, weather conditions, satellite states, and so on.[3] For example, the arrival times and count of tasks submitted by users are uncertain, which intensifies the difficulty for imaging satellite resource management in emergency. Thereby, providing a novel planning and scheduling algorithm for emergency tasks is mandatory due to the critical natures of tasks and dynamic environment.

Up to now, a great deal of scheduling algorithms of imaging satellites has been developed (see details in Related Work). Unfortunately, to the best of our knowledge, less work has been done with respect to dynamic emergency scheduling of multiple imaging satellites. The difficulties for this issue are three folds as follows:

1) The features of time windows, users' expected finish times, and dynamic environment have to be considered, which makes task scheduling on multiple satellites more complicated than traditional schemes.

2) Dynamic emergency scheduling has no fixed horizon (e.g. one day for daily planning, which is shown in Ref. 4), making the scheduling of imaging satellites difficult in modeling and solving.

3) Multiple conflicting objectives need to be considered while scheduling, such as schedulability, user

---

[1] Ph.D candidate, Science and Technology on Information Systems Engineering Laboratory, No.47 Yanzheng Street, AIAA non-member

[2] Ph.D, Science and Technology on Information Systems Engineering Laboratory, No.47 Yanzheng Street, AIAA non-member

[3] Professor, Science and Technology on Information Systems Engineering Laboratory, No.47 Yanzheng Street, AIAA non-member

satisfaction ratio and stability, etc.

The aforementioned arguments present a big challenge to design and implement novel and fast dynamic scheduling algorithms for emergency tasks submitted to imaging satellites, which motives us to develop an efficient dynamic emergency scheduling strategy to solve this issue.

1) We established a multi-objective mathematic programming model to formulate the dynamic emergency scheduling problem of multiple imaging satellites.

2) Moreover, we proposed a task dynamic merging (DM) policy to enhance the satellite resource utilization.

3) With the dynamic merging in place, we designed a novel dynamic emergency scheduling (DM-DES) algorithm of multiple imaging satellites.

4) By extensive simulation experiments, we discovered that the DM-DES algorithm could efficiently improve the scheduling quality of conventional scheduling algorithms.

The reminder of the paper is organized as follows. The next section reviews previous work in literature. Section Ⅲ formally models the dynamic emergency scheduling problem. The next two sections describe the task dynamic merging policy, as well as dynamic emergency scheduling algorithm—DM-DES. This is followed by extensive simulation experiments and performance analysis in Section Ⅵ. Finally, Section Ⅶ concludes the paper with a summary.

II. Related Work

Over the past decades, a great deal of studies have been developed on the scheduling of imaging satellites, most of which are focused on static scheme, i.e., making scheduling decisions in an off-line planning phase. Bensana et al. investigated the daily management problem of an imaging satellite SPOT-5, and formulated the problem as a Variable Valued Constraint Satisfaction Problem or as an Integer Linear Programming Problem. Besides, several exact methods like Depth First Branch and Bound or Russian Dolls search were proposed to find the optimal solution, as well as approximate methods like Greedy search or Tabu search to find a good solution.[4] Cordeau et al. described the problems selecting a subset of requests for each orbit yielding a maximum profit within constraints, i.e., satellite orbit problem (SOP), and presented a Tabu search heuristic.[5] Again, to solve the scheduling problem of imaging satellite, Lin et al. employed the Lagrange Relaxation technique to integrate with other approaches, like Tabu search or Liner search. [6, 7] In addition, Bianchessi et al. described an improved Tabu search algorithm to solve the multi-satellite, multi-orbit and multi-user scheduling.[1] Globus et al. analyzed and discussed the multi-satellite scheduling issue, and developed an evolutionary algorithm to solve the problem. Moreover, they compared the evolutionary algorithm with other existing algorithms including hill climbing (HC), simulated annealing (SA), two variants of genetic algorithm, etc.[8, 9] It should be noted that the aforementioned static scheduling methods have definite horizons. Once a scheduling decision is made, it cannot be changed, which is obviously not feasible in dynamic environment.

There also exist a few attentions directed towards the dynamic scheduling of imaging satellites. Pemberton and Greenwald discussed the dynamic scheduling problem and analyzed the contingency conditions.[3] Besides, central to max-flexibility retraction heuristic, Kramer and Smith suggested a repair-based search method for oversubscribed scheduling problem.[10] With the similar idea, Wang et al. proposed a heuristic to solve dynamic scheduling problem focused on multiple imaging satellites.[11] Unfortunately, all the above dynamic scheduling schemes only focus on tasks with no timing requirements, lacking of guarantee to emergency tasks within their expected finish times.

For task merging, only Cohen et al. considered the context of multiple targets in a single scene, viewed as a preliminary investigation of task merging.[12]

In this paper, we concentrate on designing a novel task dynamic merging policy, as well as applying it in our proposed dynamic emergency scheduling algorithm of multiple imaging satellites.

III. Problem Formulation

Contrasted with static scheduling, dynamic scheduling mainly handles aperiodic tasks whose arrival times are not known a priori, which is also the most important feature of emergency tasks besides users' expected finish times. Without loss of generality, we formulate the dynamic emergency scheduling of multiple imaging satellites with a multi-objective mathematic programming model that contains five basic objects, i.e., tasks, resources, available opportunities, operational constraints and objectives.

Tasks. In this paper, we focus on dealing with targets, which can be photographed in a scene of a sensor. A set $T = \{t_1, t_2, ..., t_n\}$ of emergency tasks are dynamically submitted for execution. Each task $t_i \in T$ has a priority p_i, an arrival time a_i and an expected finish time e_i. Considering the practical condition that many tasks are submitted together, thus, we assume the new tasks arrive in batch style.

Resources. A set $R = \{r_1, r_2, ..., r_m\}$ of resources (or sensors) are available for assignment to tasks. Each resource $r_j \in R$ is denoted by $r_j = (d_j, \sigma_j, s_j, b_j, o_j, as_j, msg_j)$, where $d_j, \sigma_j, s_j, b_j, o_j, as_j$ and msg_j are the duration of task execution, field of view, slewing pace, start-up time, retention time of shutdown, attitude stability time and maximum slewing angle, respectively. Since each target is imaged in a scene and the size can be neglected, all tasks on each resource $r_j \in R$ have the same duration denoted by a small constant d_j.

Available Opportunities. We denote $AO_{ij} = \{ao_{ij1}, ao_{ij2}, ..., ao_{ijK_{ij}}\}$ as a set of available opportunities of task t_i on resource r_j. For a given available opportunity $ao_{ijk} \in AO_{ij}$, it is represented by $ao_{ijk} = \{[ws_{ijk}, we_{ijk}], \theta_{ijk}\}$, where $[ws_{ijk}, we_{ijk}]$ is the time window and θ_{ijk} is the ideal slewing angle, which is depicted in Figure 1.

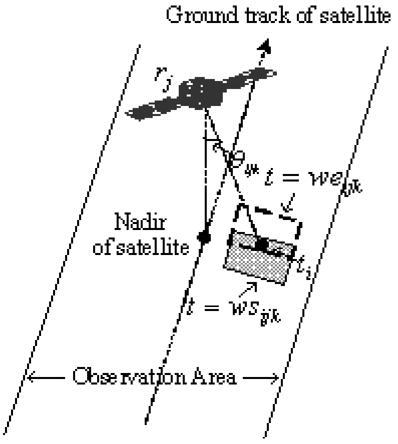


Figure 1. Available Opportunity ao_{ijk}

Based on the expected finish time of t_i, we divided AO_{ij} into three subsets, i.e., valid available opportunity set $v\text{-}AO_{ij}$, critical available opportunity set $c\text{-}AO_{ij}$ and delay available opportunity set $d\text{-}AO_{ij}$. For any available opportunity ao_{ijk}, 1) if $we_{ijk} \le e_i$, $ao_{ijk} \in v\text{-}AO_{ij}$; 2) if $ws_{ijk} \le e_i < we_{ijk}$, $ao_{ijk} \in c\text{-}AO_{ij}$; 3) if $ws_{ijk} > e_i$, $ao_{ijk} \in d\text{-}AO_{ij}$. It is defined that the amounts of elements in AO_{ij}, $v\text{-}AO_{ij}$, $c\text{-}AO_{ij}$ and $d\text{-}AO_{ij}$ are $K_{ij}, K_{ij}^v, K_{ij}^c$ and K_{ij}^d, respectively. Apparently, K_{ij}^v is a finite integer, K_{ij}^c is equal to "1" or "0", and K_{ij}^d is an infinite integer. In practice, tasks cannot wait to be executed for a long time (e.g., several days), hence we assume there exists a due date dd_i for each task t_i in T, and $ws_{ijk} < dd_i$. As a result, K_{ij}^d is also a finite integer, as shown in Figure 2, $K_{ij} = K_{ij}^v + K_{ij}^c + K_{ij}^d$.

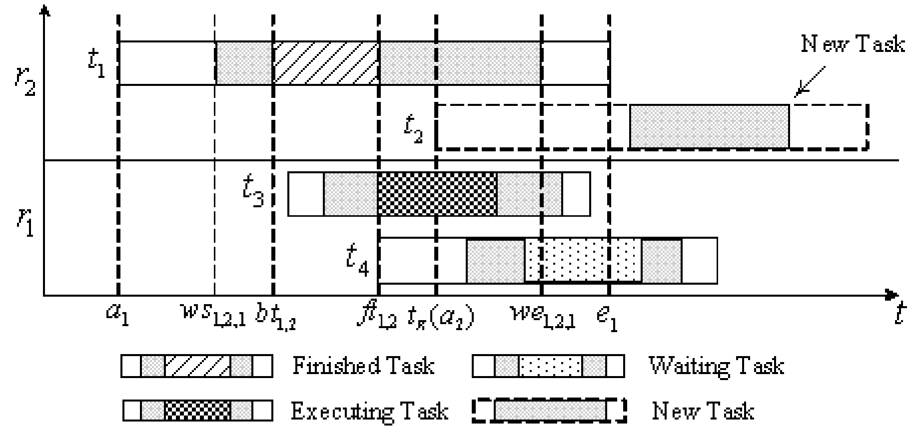


Figure 3. Four Sorts of Tasks While Dynamic Scheduling. *Regarding task t_1, a_1 and e_1 represent the arrival time and expected finish time of t_1, respectively; $bt_{1,2}$ and $ft_{1,2}$ denote the beginning time and finish time of t_1 on resource r_2; as well as $ws_{1,2,1}$ and $we_{1,2,1}$ are the start time and end time of the time window in the 1st available opportunity $ao_{1,2,1}$ of t_1 on r_2, respectively. Task t_1 is allocated to the 1st available opportunity on resource r_2, $x_{1,2,1} = 1$. Labels for other tasks are omitted for simplicity.*

An allocation matrix $X = (x_{ijk})_{n \times m \times K_{ij}}$ is used to reflect a mapping of tasks, where element x_{ijk} equals to "1" if task t_i is allocated to the kth available opportunity on resource r_j; otherwise, x_{ijk} is "0". Additionally, bt_{ij}, ft_{ij} and

ϕ_{ij} represent the beginning time, finish time and observation angle of task t_i on resource r_j, respectively, where $ft_{ij} = bt_{ij} + d_j$.

For facilitating understanding the dynamic emergency scheduling of imaging satellites, we define in our study four sorts of tasks: finished tasks (FT), executing tasks (ET), waiting tasks (WT) and new tasks (NT), which are illustrated in Figure 3.

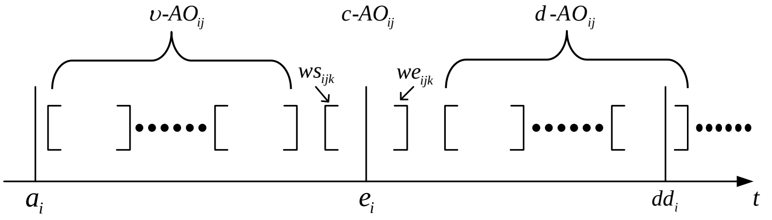


Figure 2. Available Opportunity Sets

The sort of a task is determined by the timing instant t_R when a scheduling event is triggered. For any task t_i, 1) if $ft_{ix} < t_R$, $t_i \in FT$; 2) if $bt_{ix} < t_R < ft_{ix}$, $t_i \in ET$; 3) $bt_{ix} > t_R$, $t_i \in WT$; 4) if $a_i = t_R$, $t_i \in NT$. Since the non-preemptive scheme is employed in our study, the scheduling decisions for finished tasks and executing tasks cannot be changed. However, the waiting tasks and newly arrived tasks can be taken into account together for scheduling optimization. Therefore, we consider dealing with tasks in WT and NT.

Operational Constraints. Since each task is neither disjunctive nor preemptive, a task can only be allocated to one resource, and be executed once. Thereby, we have the following constraint C_1:

$$C_1: \quad \sum_{j=1}^{m} \sum_{k=1}^{K_{ij}} x_{ijk} \le 1 \qquad \forall t_i \in T \tag{1}$$

Each task t_i must be executed in an available opportunity ao_{ijk}, $ao_{ijk} \in \bigcup_{r_j \in R} AO_{ij}$. Hence, we have the available opportunity constraint as follows:

$$\begin{aligned} & \forall t_i \in T, r_j \in R, ao_{ijk} \in AO_{ij}, \text{if } x_{ijk} = 1 \\ C_2: \quad & \begin{cases} bt_{ij} \in [ws_{ijk}, we_{ijk} - d_j] \\ \varphi_{ij} \in [\max\{\theta_{ijk} - \sigma/2, -msg_j\}, \min\{\theta_{ijk} + \sigma/2, msg_j\}] \end{cases} \end{aligned} \tag{2}$$

We consider a set $T_j = \{t_{1j}, t_{2j}, \cdots, t_{Hj}\}$ ($T_j \in T \setminus FT$) of tasks waiting for execution with order on resource r_j, i.e., task t_{1j} will be executed first and t_{Hj} the last. If task t_{ij} is finished successfully, resource r_j requires sufficient setup time to prepare for executing the next task $t_{i+1,j}$. Thus, the ready time constraint must be considered. Now, we first introduce some relevant definitions.

Definition 1. Setup Time

Setup time, say $c_{i,i+1,j}$, is the minimal needed duration from the finish time of t_{ij} to the beginning time of the next task $t_{i+1,j}$ in T_j, which is denoted as below:

$$c_{i,i+1,j} = b_j + o_j + as_j + |\phi_{i+1,j} - \phi_{ij}| / s_j \tag{3}$$

where ϕ_{ij} and $\phi_{i+1,j}$ are the observation angles of tasks t_{ij} and $t_{i+1,j}$, respectively.

Definition 2. Ready Time

Ready time, say rt_{ij}, represents the time from which t_{ij} can be executed on r_j, which is defined as follows:

$$rt_{ij} = ft_{i-1,j} + c_{i-1,i,j} \tag{4}$$

Therefore, we can get the ready time constraint formulation depicted as bellow:

$$C_3: \forall t_{ij} \in T_j, r_j \in R, \; rt_{ij} \le bt_{ij} \tag{5}$$

Objectives. In this paper, we give first priority to scheduling benefit, thus, the primary objective is to maximize the sum of priorities of accepted tasks under constraints:

$$\max_{t_i \in T, r_j \in R} \{\sum_{j=1}^{m} \sum_{i=1}^{n} \sum_{k=1}^{K_{ij}} p_i x_{ijk}\} \tag{6}$$

Moreover, to make the scheduling stable, the perturbation of the whole tasks should be minimized. Before introducing this objective, we firstly define the perturbation in our study.

Definition 3. Perturbation

Perturbation δ_u is the measurement of the distance between a new schedule and the initial one in the uth scheduling.

Generally, the distance results from the following three types of variances of tasks: 1) Variance of finish time within expected finish time. 2) Variance of finish time resulting in delay, i.e., dissatisfaction of expected finish time but not being rejected. 3) Rejection.

Assume there are S batches of tasks in total. Because the scheduling event is triggered by the arrival of a batch of tasks, the scheduling times is equal to S. Thus,

$$\delta = \sum_{u=1}^{s} \delta_u = \sum_{u=1}^{s} \sum_{i=1}^{n} \sum_{v=1}^{3} \omega_v disturb_v(i,u) \tag{7}$$

where δ is the total perturbation in the scheduling of the whole tasks, $\omega_v, v = 1,2,3$ represent the influence measurement for users of type v variance, and generally $\omega_1 < \omega_2 < \omega_3$.

$$disturb_v(i,u) = \begin{cases} 1, \text{ if type } v \text{ variance happens on task } t_i \text{ at the } u\text{th scheduling} \\ 0, \text{ otherwise} \end{cases} \tag{8}$$

Consequently, the minimum perturbation objective is depicted as below:

$$\min \sum_{u=1}^{s} \sum_{i=1}^{n} \sum_{v=1}^{3} \omega_v disturb_v(i,u) \tag{9}$$

Finally, we prefer more tasks can be finished within their expected finish times, thus we have the following objective:

$$\max_{t_i \in T, r_j \in R} \{ \sum_{j=1}^{m} \sum_{i=1}^{n} \sum_{k=1}^{K_{ij}^{v}} x_{ijk} / \sum_{j=1}^{m} \sum_{i=1}^{n} \sum_{k=1}^{K_{ij}} x_{ijk} \} \tag{10}$$

Accordingly, the scheduling problem in our study can be formulated as a multi-objective mathematic programming problem with aforementioned constraints.

IV. Dynamic Merging

Since a scene imaged by a sensor is with a certain size, some adjacent targets in the same scene can be observed simultaneously, namely simultaneous observation, as shown in Figure 4a. Our dynamic merging refers to selecting a waiting task $t_{i'j}$ in T_j to merge with a new task t_i for simultaneous observation, generating a new waiting task t_{i^*j}. Assume $t_{i'j}$ is allocated to its lth available opportunity on r_j, then the dynamic merging must satisfy the following constraints:

Available resource constraint:

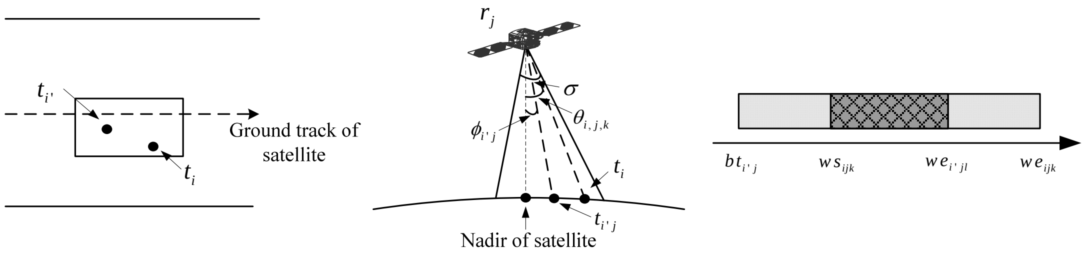
$$K_{ij} > 0 \tag{11}$$

Visibility constraints:

$$\exists k \in [1,...,K_{ij}]$$
$$\text{s.t.} \begin{cases} (a): \quad \theta_{ijk} \in [\phi_{i'j} - \sigma_j / 2, \phi_{i'j} + \sigma_j / 2] \\ (b): \ \min\{we_{ijk}, we_{i'jl}\} - \max\{bt_{i'j}, ws_{ijk}\} \ge d_j \\ (c): \ \max\{bt_{i'j}, ws_{ijk}\} + d_j + c_{i',i'+1,j} \le bt_{i'+1,j} \end{cases} \tag{12}$$

Available resource constraint means that the resource r_j must be available for the new task t_i. With respect to visibility constraints, we can get:

- (a) is the slewing angle constraint. As illustrated in Figure 4b, the target of task t_i must be in the observation scene of resource r_j when executing task $t_{i'j}$. Because the observation angle of $t_{i'j}$ is immutable, due to not resulting in conflicts with other tasks in T_j.



(a) (b) (c)

Figure 4. Dynamic merging Constraints

- (b) is the time window constraint. It is shown in Figure 4c that tasks t_i and $t_{i'j}$ must be executed satisfying their own available opportunity constraints, respectively, and the beginning time cannot be brought forward, due to not conflicting with the former tasks.
- (c) is the setup time constraint. It is implied that the setup time should be enough to execute the next task after merging.

Besides, the beginning time of the new waiting task t_{i^*j} is $\max\{bt_{i'j}, ws_{ijk}\}$, the finish time $ft_{i^*j} = bt_{i^*j} + d_j$, the observation angle $\phi_{i^*j} = \phi_{i'j}$. Moreover, the priority of task t_{i^*j} is $p_{i^*} = p_i + p_{i'}$, and the expected finish time $e_{i^*} = \min\{e_i, e_{i'}\}$.

All the tasks in WT satisfying the merging constraints with t_i compose an alternative task set ATS_i for dynamic merging. The role of the following alternative task set establishment (ATSE) algorithm is to establish the set ATS_i.

Algorithm 1 ATSE Algorithm

```
Initialize the set $ATS_i = \phi$, available opportunity set $AO_i = \bigcup_{r_j \in R} AO_{ij}$;
for each available opportunity $ao_{ijk}$ in $AO_i$, do
    for each task $t_{i'j}$ in $T_j$, do
        if $t_{i'j}$ and $ao_{ijk}$ satisfy the constraints (11) ~ (12) then
            $ATS_i = ATS_i \bigcup t_{i'j}$;
        end if
    end for
end for
```

It is denoted in **Algorithm 1** that ATSE firstly explores all the available opportunities of t_i on the whole resources in R. For each available opportunity ao_{ijk}, ATSE seeks after the whole tasks in T_j which can be merged with t_i from ao_{ijk}, and then put them into the alternative task set ATS_i. Finally, ATS_i contains all tasks that can be merged with t_i.

V. Emergency Scheduling Algorithm

The dynamic emergency scheduling of multiple imaging satellites is a NP-complete problem,[3] which motivates us to use heuristic approaches to allocate emergency tasks for a close-to-optimal solution. In our study, the dynamic scheduling can be viewed as task insertions from NT to the current schedule.

A. Task Insertion Conditions

Task insertion is to insert a task from NT into the waiting task set WT within the operational constraints. For fear of the complicated constraint checking, we give the straightforward task insertion conditions in this section.

A new task is usually inserted into the idle time slot between two tasks in $T_j, j \in [1,...,m]$. Also, a special case is that the new task is inserted before the first task or after the last task in T_j. Without loss of generality, we add two virtual tasks, i.e., a primal task t_{sj} and a terminative task t_{ej} in each T_j with the following properties: 1) The beginning time of t_{sj} is t_R, which is the timing instant of scheduling, ft_{sj} is the finish time of the task that is being executed at t_R. If there is no task being executed, $ft_{sj} = t_R$; 2) Both the beginning time and finish time of task t_{ej} are ∞; 3) The setup times of t_{sj} with other tasks are 0, so as t_{ej}.

Based on the above construction, the task insertion can be described as inserting a task into an idle time slot of r_j between task $t_{i'j}$ and $t_{i'+1,j}$, denoted by $< r_j, t_{i'j}, t_{i'+1,j} >$. The insertion conditions are depicted as below:

Available resource condition:

$$K_{ij} > 0 \tag{13}$$

Time conditions:

$$\exists k \in [1,...,K_{ij}]$$
$$\text{s.t.}\begin{cases}(a):\ rt_{ij} \le we_{ijk} - d_j \\ (b):\ rt_{i'+1,j} \le bt_{i'+1,j}\end{cases} \tag{14}$$

The available resource condition indicates that the resource r_j must be available for t_i. As for the time conditions:

- (a) denotes that the beginning time of task t_i must satisfy the available opportunity constraint C_2 in Section III;
- (b) represents that the insertion of the new task t_i cannot affect the execution of the subsequent tasks.

If task t_i satisfies the conditions above, it can be inserted into $< r_j, t_{i'j}, t_{i'+1,j} >$, to yield a new waiting task t_{ij} with the beginning time $bt_{ij} = \max\{rt_{ij}, ws_{ijk}\}$, the finish time $ft_{ij} = bt_{ij} + d_j$, and the observation angle $\phi_{ij} = \theta_{ijk}$.

B. Some Rules

There are two steps that are very critical in the dynamic emergency scheduling of multiple imaging satellites, i.e., selecting new tasks and selecting tasks for dynamic merging. To address this issue, we develop two corresponding rules: task requirement degree and optimal task merging.

Task Requirement Degree

Task requirement degree TRD_i represents the urgency to be allocated of t_i.

$$TRD_i = p_i / \sum_{r_j \in R} K_{ij} \tag{15}$$

Thus, a task with higher priority and less available opportunities should be allocated preferentially. This rule is for selecting new tasks in NT.

Optimal Task Merging

Definition 4. Best Merging Task, Non-perturbation Merging Task and Non-delay Merging Task

For a given task $t_i, t_i \in NT$, if $\exists t_{i'j}, t_{i'j} \in ATS_i$, that is able to be merged with t_i to generate a new waiting task t_{i^*j}.

- If $ft_{i^*j} = ft_{i'j}$, and $ft_{i^*j} \le et_{i^*}$, the task $t_{i'j}$ is called a best merging task of t_i.
- If $ft_{i^*j} = ft_{i'j}$, and $ft_{i^*j} > et_{i^*}$, the task $t_{i'j}$ is called a non-perturbation merging task of t_i.
- If $ft_{i^*j} \ge ft_{i'j}$, and $ft_{i^*j} \le et_{i^*}$, the task $t_{i'j}$ is called a non-delay merging task of t_i.

From Definition 4, it is easy to find that if task t_i can be merged with a best merging task, there will be neither perturbation nor delay. Thus, we should firstly select a best merging task for dynamic merging if exists. If there exist multiple best merging tasks in ATS_i, we select one randomly. If there is no best merging task in ATS_i, we arbitrarily choose a non-perturbation merging task. If no non-perturbation merging task exists, either, a non-delay merging task with minimal $ft_{i^*j} - ft_{i'j}$ will be selected. At last, if all the above tasks do not exist in ATS_i and $ATS_i \ne \phi$, we select one task with minimal $ft_{i^*j} - et_{i'j}$.

C. Algorithm Description

In this section, we present the novel dynamic emergency scheduling algorithm with dynamic merging (i.e., DM-DES) of multiple imaging satellites. The pseudocode of DM-DES is shown in **Algorithm 2**.

Algorithm 2 DM-DES Algorithm

```
1  Initialize the set $T, R,$ and $AO_{ij}$;
2  while $NT \neq \phi$ do
3      Select task $t_i$ in $NT$ with maximum $TRD_i$, $NT \leftarrow NT \setminus t_i$;
4      Establish $ATS_i$ for $t_i$ with ATSE Algorithm;
5      if $ATS_i \neq \phi$ then
6          Select one task in $ATS_i$, say $t_{i'j}$, according to optimal task merging rules;
7          Merge task $t_i$ and $t_{i'j}$ to generate a new waiting task $t_{i^*j}$;
8          break;
9      end if
10     $AO_i \leftarrow \bigcup_{r_j \in R} AO_{ij}$;
11     while $AO_i \neq \phi$ do
12         Select an available opportunity $ao_{ijk}$ with minimum $we_{ijk}$, $AO_i \leftarrow AO_i \setminus ao_{ijk}$;
13         for each task $t_{i'j}$ in $T_j$ do
14             if $ao_{ijk}$ satisfies the conditions (13) ~ (14) with $<r_j, t_{i'j}, t_{i'+1,j}>$ then
15                 Insert the task $t_i$ into $<r_j, t_{i'j}, t_{i'+1,j}>$;
16                 Go to line 2;
17             end if
18         end for
19     end while
20 end while
```

The DM-DES algorithm firstly picks out a new task t_i from NT according to task requirement degree (line 3). Then the DM-DES calls ATSE algorithm, to get all the tasks in WT which can be merged with t_i (line 4).

If the alternative task set ATS_i is not empty, i.e., some tasks in WT can be merged with t_i, DM-DES chooses one task from ATS_i in terms of optimal task merging rule to merge with t_i, which implies that t_i has been allocated successfully, and then DM-DES goes back to line 2 to select another task until no tasks are in NT (lines 5~9).

If no tasks are capable of being merged with t_i, DM-DES explores the available opportunities of t_i on the whole resources in time order. For an available opportunity ao_{ijk}, DM-DES seeks after the available time slots. If there exists an available time slot $< r_j, t_{i'j}, t_{i'+1,j} >$, which represents that t_i can be scheduled successfully; DM-DES inserts t_i into $< r_j, t_{i'j}, t_{i'+1,j} >$, and goes back to line 2 to select another task. Otherwise, if there exists no available time slots, DM-DES explores the next available opportunity. After the while loop, if the task t_i has not been inserted into a time slot, it is denoted that t_i fails to be scheduled. Then DM-DES goes back to line 2 to allocate another task until no tasks are in NT (lines 11~19).

VI. Performance Evaluation

We evaluate in this section the performance of the proposed DM-DES algorithm. To demonstrate the performance improvements of DM-DES, we quantitatively compare it with RBHA which was proposed in Ref. 11, and a baseline algorithm—dynamic emergency scheduling algorithm without dynamic merging (DES).

1) RBHA: The basic idea of RBHA is to carry out some amount of iterative repair search inside each new task's available opportunities. Within the repair search for a given new task $t_i \in NT$, some rules are adopted to decide which tasks to "temporarily" retract, making room for incorporating t_i.

2) DES: DES is a variant of the DM-DES algorithm. The difference between DES and DM-DES is that DES does not consider dynamic merging. The goal of introducing DES is to evaluate the effectiveness of task dynamic merging.

To make the comparison fair, we slightly modify the RBHA in such a way that it chooses the available opportunities in $\bigcup_{r_j \in R} v\text{-}AO_{ij}$ firstly. If no available opportunity is obtained, we then select the other available opportunities.

The performance metrics by which we evaluate the system performance include the following:

1) Total Task Priorities (TTP). This is defined as: $TTP = \sum_{j=1}^{m}\sum_{i=1}^{n}\sum_{k=1}^{K_{ij}} p_i x_{ijk}$.

2) Satisfaction Ratio (SR). This is defined as: $SR =$ Total amount of tasks within their expected finish times/Total amount of tasks being executed.

3) Perturbation measurement δ is used to measure the scheduling stability.

A. Simulation Method and Parameters

To validate the performance improvements of DM-DES, the targets are randomly generated in the area: latitude $-30^\circ \sim 60^\circ$ and longitude $0^\circ \sim 150^\circ$. The amounts of targets are 200, 400, 600, 800, 1000 and 1200, respectively. Without loss of generality, the priorities of tasks are uniformly distributed in [1, 10]. According to extensive literatures,[3,4,5,12] three sensors on different satellites are considered in this paper. The parameters of sensors are presented in Table 1, and the orbit models of satellites are obtained from STK, where the parameters with (*) denote the designed values based on literatures.

Table 1. Parameters of sensors

Sensor	Satellite	msg (deg)	FOV	d^*(s)	s^* (deg/s)	b^*(s)	o^*(s)	as^*(s)	Height(km)
Sensor1	IKONOS-2	45	0.931	2	1	3	3	5	681
Sensor2	QuickBird-2	25	2.1	2	1	3	3	5	450
Sensor3	SPOT-5	27	2.09	2	1	3	3	5	822

1) The arrival rate of a batch is denoted as $a_i = a_{i-1} + intervalTime$, where $intervalTime$ is a random positive real number, uniformly distributed in $[a,b]$.

2) The expected finish time e_i of task t_i is chosen as follows: $e_i = a_i + Time$, where $Time$ subjects to normal distribution, $Time \sim N(baseTime, baseTime/10)$.

3) The due date dd_i of task t_i is described as $dd_i = a_i + DueDate$, where $DueDate \sim N(baseDueDate, baseDueDate/10)$.

Table 2 gives the simulation parameters and their values. In addition, regarding minimal perturbation objective, the parameters $\omega_1, \omega_2, \omega_3$ are defined as $\omega_1 = 0.5, \omega_2 = 1, \omega_3 = 2$, respectively.

Table 2. Parameters for simulation studies

Parameter	Value(Fixed)-(Varied)
Task Count	(800)-(200,400,600,800,1000,1200)
Priorities	([1,10])
Resource Count	(3)
IntervalTime(h)	([0,4])-([0,4],[4,8],[8,12])
baseTime(h)	(6)-(3,6,9,12)
baseDueDate(h)	(24)

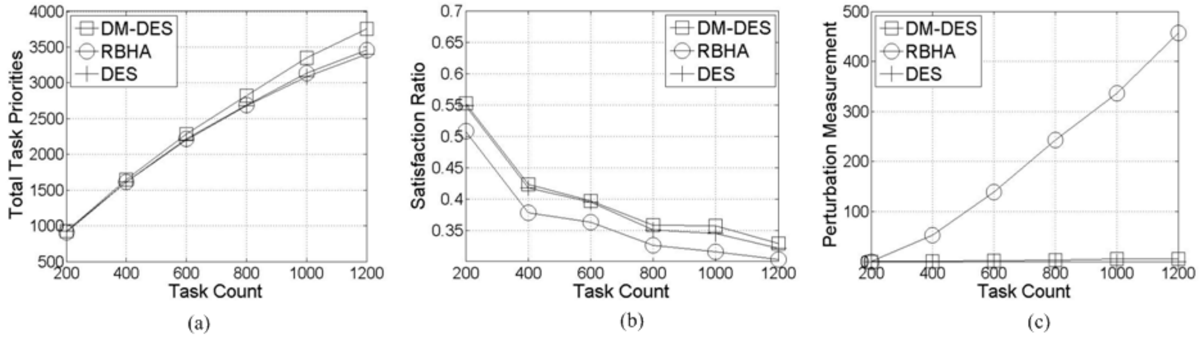


Figure 5. Performance impact of task count.

B. Performance Impact of Task Count

In this experiment, we investigate the performance impact of task count. Figure 5 shows the comparisons of DM-DES, RBHA and DES in terms of total task priorities, satisfaction ratio and perturbation measurement.

Figure 5a shows that with the increase of task count, the total task priorities of all algorithms get increased. This is because in the condition of sufficient resource capacity, the amount of accepted tasks will be increased with the increment of coming task count. In addition, DM-DES shows better scheduling quality than RBHA and DES. This confirms that DM-DES benefits from task dynamic merging that reduces conflicts, thus the scheduling quality is improved. Besides, RBHA slightly outperforms DES, because RBHA considers "temporarily" retracting some allocated tasks to accept some higher priority tasks.

From Figure 5b, it is found that the satisfaction ratios of all algorithms get decreased with the task count's increasing. This result can be attributed to the fact that the intervals between arrival times and expected finish times are finite. When the task count is getting increased, more tasks cannot be accepted within their expected finish times, making the satisfaction ratios decrease. In addition, the satisfaction ratios of both DM-DES and DES are higher than that of RBHA, which proves that task retraction may result in more tasks not being accepted within their expected finish times.

Based on Figure 5c, owing to task retraction strategy, RBHA may make the schedule greatly change, even some accepted tasks be rejected in rescheduling, thus RBHA performs the highest perturbation measurement. With respect to DM-DES, perturbation can only be induced by dynamic merging; hence, the perturbation measurement of DM-DES is extremely less than that of RBHA. Besides, the perturbation measurement of DES always keeps "0", because it considers neither dynamic merging nor retraction, i.e., the allocated tasks cannot be adjusted in rescheduling. In addition, the perturbation measurements of both DM-DES and RBHA get larger with the increase of task count.

C. Performance Impact of Arrival Rate

We carry out a group of experiments in this section to observe the impact of arrival rate on all algorithms. The experimental results are depicted in Figure 6.

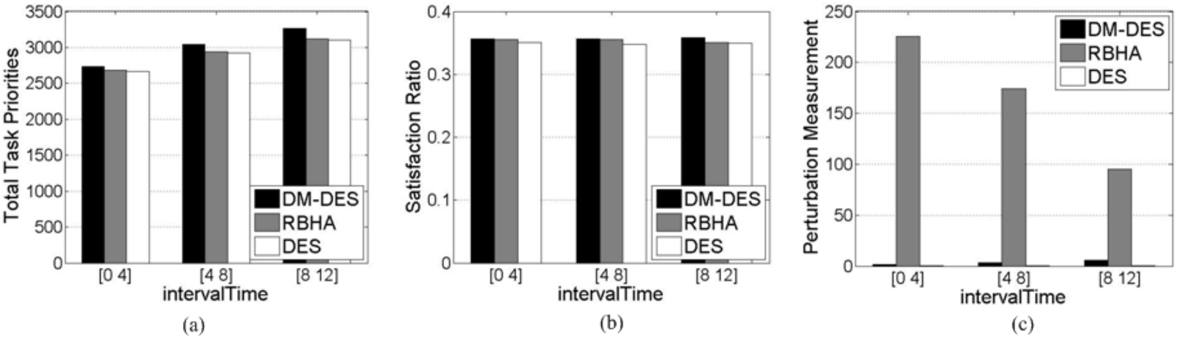


Figure 6. Performance impact of arrival rate.

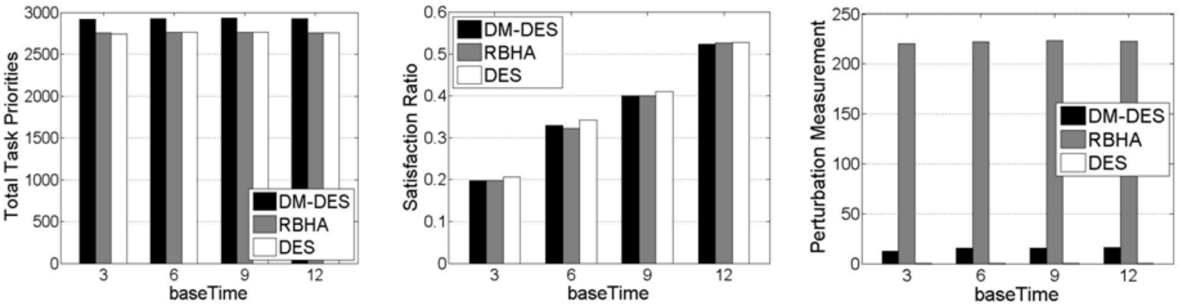
Figure 6a shows that with the increase of *intervalTime*, the total task priorities of all algorithms increase. The reason is that the arrival rate of emergency tasks becomes less with the increase of *intervalTime*, thus less tasks are waiting on resources, leading to tasks having less conflicts. Hence, more tasks will be accommodated. Besides, DM-DES is superior to the other algorithms in terms of schedulability as the reasons described in the previous sections.

From Figure 6b, it is found that the satisfaction ratios have no obvious variances. This confirms that both the counts of tasks accommodated within expected finish times and not increase with the increase of *intervalTime*, thus, the satisfaction ratios keep invariable basically.

Figure 6c depicts that RBHA always has the highest perturbation measurement, and DES always keeps "0", as the explanations in Figure 5c. Moreover, the perturbation measurement of RBHA descends with the reduction of the arrival rate, which can be explained by that with less arrival rate, the system load becomes lighter, and fewer tasks will be adjusted in rescheduling. Oppositely, the perturbation measurement of DM-DES increases, which proves that less arrival rate may induce more opportunities for dynamic merging.

D. Performance Impact of Expected Finish Time

The objective of this experiment is to investigate the impact of task expected finish time on the performance of the three algorithms. Figure 7 illustrates the experimental results.



(a) (b) (c)

Figure 7. Performance impact of expected finish time.

The results in Figure 7a show that the total task priorities of the algorithms have no obvious change when the expected finish times become looser. This can be attributed to the fact that the total amount of accepted tasks will not change because the due dates of tasks are invariable. The increasing of expected finish times enhances the amount of tasks accommodated within their expected finish times but reduces the amount of the others.

Figure 7b illustrates that the satisfaction ratios of the algorithms keep ascending trends with the increase of expected finish times. The reason is that the increase makes more tasks be accepted within their expected finish times, but the total amount of accepted tasks is invariable. Hence, the satisfaction ratios increase as the expected finish times go up.

Figure 7c shows that the perturbation measurements of the algorithms have no obvious variances. Because the variances of expected finish times just affect the count of accepted tasks within their expected finish times, having no obvious influences on the other metrics.

VII. Conclusion

This paper establishes a multi-objective mathematic model to formulate the dynamic emergency scheduling problem of multiple imaging satellites. Besides, we propose a novel task dynamic merging strategy, and develop an alternative task set establishment (ATSE) algorithm used for dynamic merging. Moreover, we present a novel dynamic emergency scheduling algorithm DM-DES, which adopts dynamic merging to improve the scheduling quality in terms of schedulability, satisfaction ratio and stability. We conduct extensive experiments to compare it with its variant DES as well as RBHA.[11] The experimental results prove that the DM-DES is an excellent scheduling algorithm and is suitable for dynamic emergency scheduling of imaging satellites.

Acknowledgments

This research was supported by the National Natural Science Foundation of China No. 61104180 and the National High-Tech Research and Development Plan of China under Grant No. 2008AA7070412.

References

[1]Bianchessi, N., Cordeau, J., Desrosiers, J., Laporte, G., and Raymond, V., "A heuristic for the Multi-satellite, Multi-orbit and Multi-user Management of Earth Observation Satellites," *European Journal of Operational Research*, Vol. 177, 1 Mar. 2007, pp. 750-762.

[2]Zhu, K. J., Li, J. F., and Baoyin, H. X., "Satellite Scheduling Considering Maximum Observation Coverage Time and Minimum Orbital Transfer Fuel Cost," *Acta Astronautica*, Vol. 66, 2010, pp. 220–229.

[3]Pemberton, J., and Greenwald, L., "On the Need for Dynamic Scheduling of Imaging Satellites," *Proceedings of the American Society for Photogrammetry and Remote Sensing*, Denver, Colorado, 2002.

[4]Bensana, E., Verfaillie, G., Agnese, J., Bataille, N., and Blumestein, D., "Exact and Inexact Methods for the Daily Management of an Earth Observation Satellite," *Proceedings of the International Symposium on Space Mission Operations and Ground Data Systems*, AIAA, Munich, Germany, 1996, pp. 507-514.

[5]Cordeau, J., and Laporte, G., "Maximizing the Value of an Earth Observation Satellite Orbit," *Journal of the Operational Research Society*, 2005, pp. 962–968.

[6]Lin,W. C., Liao, D., Liu, C., and Lee, Y., "Daily Imaging Scheduling of an Earth Observation Satellite," *IEEE Transactions on Systems, Man, Cybernetics-Part A: Systems And Humans*, 2005, pp. 213–223.

[7]Lin, W. C., and Liao, D. Y., "A Tabu Search Algorithm for Satellite Imaging Scheduling," *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, IEEE, Hague, Netherlands, 2004, pp. 1601–1606.

[8] Globus, A., Crawford, J., Lohn, J., and Pryor, A., "A Comparison of Techniques for Scheduling Fleets of Earth-Observing Satellites," *Journal of the Operational Research Society*, 2003, pp. 962–968.

[9] Globus, A., Crawford, J., Lohn, J., and Pryor, A., "Scheduling Earth Observing Satellites with Evolutionary Algorithms," *Proceedings of the Space Mission Challenges for Information Technology Conference*, IEEE, California, 2003.

[10] Kramer, L. A., and Smith, S. F., "Maximizing Flexibility: A Retraction Heuristic for Oversubscribed Scheduling Problems," *Proceedings of the eighteenth International Joint Conference on Artificial Intelligence*, Acapulco, Mexico, 2003, pp. 1218–1223.

[11] Wang, J. M., Li, J. F., and Tan, Y. J., "Study on Heuristic Algorithm for Dynamic Scheduling Problem of Earth Observation Satellites," *Proceedings of the 8th International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing*, IEEE, Qingdao, China, 2007, pp. 9–14.

[12] Cohen, R., "Automated Spacecraft Scheduling the ASTER Example," *Proceedings of the Ground System Architecture Workshop*, 2002.