

Research Article

A Dynamic Scheduling Method of Earth-Observing Satellites by Employing Rolling Horizon Strategy

Qiu Dishan, He Chuan, Liu Jin, and Ma Manhao

Science and Technology on Information Systems Engineering Laboratory, National University of Defense Technology, Changsha 410073, China

Correspondence should be addressed to He Chuan; chuanhe@nudt.edu.cn

Received 31 December 2012; Accepted 17 February 2013

Academic Editors: A. De Felice, K. Kudela, and M. Ragulskis

Copyright © 2013 Qiu Dishan et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Focused on the dynamic scheduling problem for earth-observing satellites (EOS), an integer programming model is constructed after analyzing the main constraints. The rolling horizon (RH) strategy is proposed according to the independent arriving time and deadline of the imaging tasks. This strategy is designed with a mixed triggering mode composed of periodical triggering and event triggering, and the scheduling horizon is decomposed into a series of static scheduling intervals. By optimizing the scheduling schemes in each interval, the dynamic scheduling of EOS is realized. We also propose three dynamic scheduling algorithms by the combination of the RH strategy and various heuristic algorithms. Finally, the scheduling results of different algorithms are compared and the presented methods in this paper are demonstrated to be efficient by extensive experiments.

1. Introduction

The mission of an earth-observing satellite (EOS) is to scout targets with a certain range of ground to produce high-resolution photographs [1–4]. According to the working mechanism, the imaging sensors load in EOS can be divided into visible light imaging, microwave imaging, infrared imaging, and so forth. Since they have numerous merits such as rapider response, broad coverage range, longer duration, and freedom from airspace boundaries, EOS supports many important services, such as military surveillance, geodesy and navigation, remote sensing, and monitoring.

Nowadays, EOS is attracting more and more interests worldwide be accompanied with the dramatic increase of the demand for imaging service. One major research trend is that the single satellite used in early reconnaissance is replaced by cooperation of large satellites, yielding the so-called multi-satellite application. Unfortunately, as the number of satellites grows large, the traditional manual coordination will no longer be feasible because multisatellite scheduling (MSS) is an NP-hard combinatorial optimization problem [5]. Therefore, the researches on MSS refer to assigning observation resource to match various imaging requirements as indispensable.

There exist numerous studies on scheduling algorithm for multi-satellite to realize automated resource planning. Wang et al. [6] proposed a hybrid ant colony optimization (HACO) algorithm to overcome the disadvantage of current ant colony optimization (ACO) algorithm which is more easily plunged into local optimal solution in solving MSS problem. Jian and Cheng [7] constructed an integer programming model of MSS based on the analysis of the resource constraints and task characteristic. To solve this combinatorial optimization problem, they provided a genetic particle swarm optimization (PSO) algorithm which searches only in the appointed integer space. He et al. [8] presented a cooperative scheduling architecture of multiple satellites by converting this scheduling problem into a main problem and a subproblem. In addition, an improved PSO algorithm was used to solve MSS problem by taking task benefits as an optimization objective.

However, those above researches have been primarily focused on static scheduling problem of EOSs. It is usually assumed that the imaging tasks have been submitted before scheduling and their information is acquired. In practice, the requests from customers are continuously delivered, which lead to the imaging tasks arriving one by one. The most significant feature of dynamic scheduling is time urgency; that

is, the task must be completed within a specified time limit, or it will lose its execution value caused by failure. The satellite imaging reconnaissance mission generally has a deadline, which can reflect its execution urgency. Execution of the task must be completed within the specified deadline; otherwise, the expected benefits will not be obtained. The traditional static scheduling methods tend to overlook the timeliness feature of imaging task, which makes them inapplicable to the dynamic scheduling problem of imaging satellite.

At present, there are a few works on the dynamic scheduling problem of EOSs. Baolin et al. [9] described a new satellite mission scheduling algorithm based on constraint satisfaction problem (CSP). Zhu et al. [10] considered the problems of satellite scheduling for realizing optimal disaster rescue and proposed a hybrid algorithm to solve this multiple objects optimization problem. Wang et al. [11] considered the characteristic of EOS in dynamic scheduling and proposed a rule-based heuristic algorithm to solve this problem. The study mentioned above still has the following shortages.

- (1) It is difficult for these proposed algorithms to generate a task planning within a short time. A dynamic planning process consists of repeated scheduling events, and the traditional intelligence algorithm (IA) has high timing complexity, which cannot rapidly generate the planning scheme. Therefore, the high-efficiency heuristic algorithm should be used to address the dynamic scheduling problem of EOSs.
- (2) The impacts of scheduling time on the available tasks were not considered. Since tasks are dynamic arrivals, the planning system collects dissimilar task sets at different scheduling times. Thus, the task set should be determined based on current scheduling time before the scheduling.
- (3) The constraints during dynamic scheduling have not been adequately considered. Many constraints (i.e., the storage capacity, maximum swing angle, and continuous observation time) which have been simplified in static scheduling should be considered in dynamic scheduling.

In this paper, we tackle the above challenges imposed on the dynamic scheduling problem of EOSs by handling the impacts of deadline constraint and scheduling time on planning scheme. The integer programming model is constructed based on various constraints in actual reconnaissance activities, and the rolling horizon (RH) strategy and heuristic algorithms are employed to solve this model.

The remainder of this paper is organized as follows. Section 2 describes the dynamic scheduling problem of EOSs, and establishes the integer programming model. Section 3 designs the scheduling architecture and algorithms based on the RH optimization. Simulation results and performance analysis are given in Section 4. Finally, Section 5 concludes the paper with some future research directions.

2. Problem Description and Modeling

The EOS operates in the space in a certain orbit as shown in Figure 1. The view coverage of EOS can be formed on the ground by the subsatellite point of satellite platform as well as the view angle, swing angle, and tilting angle of satellite payload [12–14].

The purpose in addressing the dynamic scheduling problem of EOSs is to appoint observation resources and execution time for the dynamical submitted tasks with various constraints, so as to maximize the task benefits of reconnaissance activity and minimize the resource consumption as far as possible.

The imaging tasks in dynamic scheduling problem are submitted to the planning system in independent times compared to the static scheduling problem which can obtain all the tasks in advance. The dynamic scheduling system only acquires the information of arrived tasks but can not gain the situation of following tasks. Hence it needs to trigger multiple scheduling in order to cope with the new tasks which are successively submitted to the scheduling system. Therefore, the dynamic scheduling algorithms designed in this paper should have the overall coordination capacity; that is, the algorithm should be able to timely adjust the execution scheme of planned tasks for executing the emergency tasks submitted later, so as to maximize task benefits. For future reference, we summarize main notations used in this paper as the following:

$T_p = [t_{\text{start}}, t_{\text{end}}]$ is the reconnaissance activity period of satellites, where t_{start} is the starting time and t_{end} is the ending time of the observation activity;

$T_S = \{t_S^0, t_S^1, \dots, t_S^p\}$ is the scheduling time set, where t_S^i ($i \geq 1$) is the i th scheduling time, $t_S^0 = t_{\text{start}}$ is the virtual initial scheduling time, $T_{\text{period}} = (t_S^p - t_S^0)/p$ is the average scheduling period, and p is the quantity of scheduling activities during the reconnaissance activity period;

Task = {task₁, task₂, ..., task_m} is the imaging task set, where task _{i} is the i th task, and m is the task number;

Sat = {sat₁, sat₂, ..., sat_n} is the satellite resource set, where sat _{j} is the j th satellite, and n is the satellite number;

$TW_i = \bigcup_{j=1}^n \bigcup_{k=1}^{q_{i,j}} tw_{i,j}^k$ is the observation opportunity window set of task _{i} between its arrival time a_i and deadline d_i , where $q_{i,j}$ is the opportunity window number of task _{i} on sat _{j} , $tw_{i,j}^k = [ws_{i,j}^k, we_{i,j}^k]$ is the k th opportunity window of task _{i} on sat _{j} , and $ws_{i,j}^k$ and $we_{i,j}^k$ are the starting time and ending time of $tw_{i,j}^k$, respectively.

In addition, task _{i} has an execution benefit p_i and it requires a continuous observation time d_i . If task _{i} can be executed before its deadline, assume ts_i is the starting time, te_i is the ending time, and h_i is the required storage capacity. Furthermore, the satellite has a swing angle gh_i and a tilting

angle gw_i to execute task $_i$. Assume a_j is the view angle of sat $_j$, β_j is the maximum swing angle, γ_j is the maximum tilting angle, ω_j is the swing speed, ρ_j is the tilting speed, tc_j is the position transfer stabilization time, r_j is the maximum position transfer frequency, and M_j is the memory capacity.

The decision of scheduling times is affected by many factors, for example, the quantity and density of task, the upload period of satellite instructions, and the communication capacity of control center. The decision variable of the dynamic scheduling problem provided in this paper is as follows:

$$x_{i,j}^k = \begin{cases} 1, & \text{if task}_i \text{ is executed in } tw_{i,j}^k, \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

Assume t_S^k is the current scheduling time and the maximum task benefits are the optimization object, then we can build the integer programming model of the dynamic scheduling problem as follows [15]:

$$\begin{aligned} \max \quad & Z = \sum_{i=1}^m \sum_{j=1}^n \sum_{k=1}^{q_{i,j}} x_{i,j}^k p_i \\ \text{s.t.} \quad & \left\{ \begin{aligned} & \sum_j \sum_k x_{i,j}^k \leq 1, \\ & [ts_i, te_i] \cap [ts_{i'}, te_{i'}] = \emptyset, \quad (1) \\ & a_i \leq t_S^k \leq ts_i, \\ & \text{if } \sum_{j=1}^n \sum_{k=1}^{q_{i,j}} x_{i,j}^k = 1, \quad (2) \\ & [ts_i, te_i] \subset \bigcup_{k=1}^{q_{i,j}} tw_{i,j}^k \subset [t_S^k, e_i], \\ & \text{if } \sum_{k=1}^{q_{i,j}} x_{i,j}^k = 1, \quad (3) \\ & te_i - ts_i \geq d_i, \\ & \text{if } \sum_{j=1}^n \sum_{k=1}^{q_{i,j}} x_{i,j}^k = 1, \quad (4) \\ & te_i + \max \left(\frac{|gh_i - gh_{i'}|}{\omega_j}, \frac{|gw_i - gw_{i'}|}{\rho_j} \right) + tc_j \leq ts_{i'}, \\ & \text{if } (te_i \leq ts_{i'}) \wedge (x_{i,j}^k = x_{i',j}^{k'}), \quad (5) \\ & \sum_i \sum_k x_{i,j}^k h_i \leq M_j, \quad (6) \\ & \sum_i \sum_k x_{i,j}^k \leq r_j, \quad (7) \\ & i, i' \in [1, 2, \dots, m], j \in [1, 2, \dots, n] \\ & k \in [1, 2, \dots, q_{i,j}], k' \in [1, 2, \dots, q_{i',j}], \end{aligned} \right. \quad (2) \end{aligned}$$

where Z is the optimization objective; constraint (1) means each task only need to be executed once, and the execution process does not involve preemptive service; constraint (2) means the scheduling system only assigns the arrived tasks and their deadline must later than current scheduling time;

constraint (3) means that if any task is executed, the execution time should between current scheduling time and its deadline; constraint (4) means that if the task can be executed, its execution time should not be shorter than the required continuous observation time; constraint (5) means any two tasks assigned to the same satellite should have an adequate time interval between their execution times to ensure the sensor of satellite can adjust its gesture; constraint (6) means that the memory store any task should not exceed its maximum capacity; constraint (7) means that the gesture transfer number of satellite in one orbit should not be higher than the allowable frequency.

3. Dynamic Scheduling Approach

3.1. Architecture of Algorithm. In this paper, the dynamic scheduling algorithm based on the RH strategy [16–18] is proposed to allocate tasks which have special characteristic such as the independent arrival time and deadline. This method can conduct scheduling (or rescheduling) forward for a task set through rolling advance, so as to effectively handle the uncertain factors brought by the dynamic tasks. The heuristic algorithms are incorporated with RH strategy to reduce the timing complexity of scheduling while each scheduling requires one replanning. In other words, the task set that needs to be scheduled is determined by the RH strategy firstly, and then the heuristic algorithms are used to assign satellites and execution periods to each task.

3.2. Rolling Horizon Strategy. The basic method of RH strategy is to divide the tasks into multiple task sets with certain overlaps based on the arrival sequence, and the division can be continuously updated along with the scheduling time. Each scheduling will decide and only assign its task set, which is called as rolling horizon. The new tasks are continuously added to the rolling horizon, and the finished tasks are gradually deleted with the advancement of the scheduling time, so as to realize the update of rolling horizon. The advantage of RH strategy is that it can decompose the complicated dynamic scheduling problem into multiple simple static scheduling sub-problems, and the optimization solution of previous problem is replaced with the optimized solutions of sub-problems, so that the complexity of the original problem will be reduced.

3.2.1. Task States. In general, tasks will go through four states based on current scheduling time: a new task, waiting task, running task, and finished task. One task may be scheduled in different time, thus the state of task is dynamic; that is, a task might be in two states in different scheduling.

In the example shown in Figure 2, t_S^k is current scheduling time. If task $_i$ has been planned during the $(k-1)$ th scheduling (i.e., $a_i \leq t_S^{k-1}$), then task $_i$ is a finished task while $te_i \leq t_S^k$ (such as task $_1$); task $_i$ is a running task if $tb_i < t_S^k \leq te_i$; task $_i$ is a waiting task if $tb_i > t_S^k$. If task $_i$ has not been planned in the $(k-1)$ th scheduling (i.e., $t_S^{k-1} \leq a_i \leq t_S^k$), then task $_i$ is a new task.

3.2.2. Rolling Horizon. The rolling horizon is used to store the tasks that need be scheduled currently. There are two key elements about rolling-horizon: the quantity and state of tasks in rolling-horizon. From the perspective of task quantity, the more the tasks fall into the rolling horizon, the stronger the capacity of scheduling system to obtain comprehensive task information is, which is important to acquire the better solution. But the timing complexity of scheduling algorithm will also be aggravated. From the perspective of task state, the rolling-horizon consists of the running tasks, waiting tasks, and new tasks generally. Among them, the processing method for the running task is an important criterion to distinguish the preemptive and nonpreemptive scheduling. The later scheduling mode is out of the interest of this paper; that is, the rolling-horizon only includes the waiting tasks and new tasks. It should be noticed that during actual scheduling, the rescheduling of waiting task will not consume any additional resources because waiting task has not been executed yet.

3.2.3. Trigger Mode. The arrangement of scheduling time is the key factor which affects the application efficiency of RH strategy, and it is mainly determined by the trigger mode of scheduling. The general trigger modes include the following types.

(i) *Event Trigger Mode.* The scheduling is triggered while the scheduling environment is changed or a manual intervention occurred, such as a new task arrived, satellite number changed, or a scheduling requirement was sent from decision-making section. The event-trigger mode is sensitive to the scheduling environment, and it can assign the emergency tasks in time. However, this mode might cause the scheduling algorithm to have a high timing complexity due to frequent scheduling, and result in hard to generate a planning scheme rapidly.

(ii) *Period Trigger Mode.* In this mode, the scheduling event will be triggered after a certain time interval, and this time interval can be a uniform constant or dynamic variable. The period-trigger mode has capability to ensure the stable frequency of scheduling, but it can not provide timely scheme for emergency tasks with high timeliness, and also has the fault that it can not adjust scheme in dealing with the change of the satellite number.

(iii) *Mixed Trigger Mode.* Mixed-trigger mode is the combination of the aforementioned two modes. It can timely allocate the emergency tasks with less time consumption, so it is an adaptive mode to be employed in the scheduling of real-time system and dynamic system.

The mixed-trigger mode is adopted in this paper, and the scheduling time for period factors and event factors is embodied in the elements belonging to T_s . The RH strategy based on the mixed-trigger mode is depicted as follows.

In Algorithm 1, each scheduling time $t_S^k \in T_S$ will trigger one rescheduling (see line 1), and the scheduling frequency during T_p is $O(P)$. The criterion described in Figure 1 is used to divide the arrived tasks into the finished task,

running tasks, waiting tasks, and new tasks based on current scheduling time (see lines 3~12); the timing complexity is $O(m)$. The tasks in rolling-horizon are sorted and assigned to the satellite resources and execution time (see line 13). The timing complexity of this step is mainly depends on the assigning algorithm; thus we assume it is $O(A)$ temporarily. It only takes $O(1)$ to execute other lines. Hence, the timing complexity of Algorithm 1 is calculated as $O(P)[O(m) + O(A)]$.

The above algorithm needs to assign satellite resources and execution time for each task, so $O(A)$ is usually much larger than $O(m)$. The timing complexity of Algorithm 1 will be dominated by $O(A)$ if the scheduling frequency (or scheduling interval) is a constant. Then, the additional timing complexity caused by RH strategy is mainly related to the scheduling frequency but not affected by other factors such as the satellite quantity. At present, the intelligence algorithms (IA) are usually used to assign satellites and execution time to tasks in the static scheduling problem. These algorithms generally require multiple iterations to realize optimization of the scheduling scheme, and have a high timing complexity. The imaging tasks in dynamic scheduling have the timeliness feature, which requires the planning scheme to satisfy the deadline possibly, and, furthermore, the planning scheme should be rapidly generated and can be fast adjusted to adapt to the change of task set by scheduling algorithm. In Algorithm 2, the heuristic algorithm is employed to match the satellites and the tasks in order to shorten the generation time of the planning scheme. Meanwhile, the RH strategy can well adjust the previous planning scheme of planned tasks based on the change of rolling-horizon. The timing complexity of the heuristic algorithm embedded in RH strategy will be analyzed in detail in the next section.

3.3. Heuristic Algorithms. This paper has proposed the heuristic algorithms AIS, DIS, and WIS based on the arrival time priority, the deadline priority, and the waiting time priority inspired by the earlier arrived time first (EAT) algorithm [19] and earlier deadline first (EDF) algorithm [20, 21] and considering the task importance.

Let $RH = \{\text{task}_{k1}, \text{task}_{k2}, \dots, \text{task}_{km}\}$ denote the rolling-horizon in the k th scheduling. The basic parameters (including the arrival time, deadline, waiting time, and importance) of all tasks are standardized as follows:

$$f(x_{k_i}) = \frac{x_{k_i} - \min_{\text{task}_{k_i} \in RH} (x_{k_i}) + 1}{\max_{\text{task}_{k_i} \in RH} (x_{k_i}) - \min_{\text{task}_{k_i} \in RH} (x_{k_i}) + 1}. \quad (3)$$

After the standardization, we record the basic parameters of task $_{k_i}$ as AS_{k_i} , DS_{k_i} , CS_{k_i} , and WS_{k_i} respectively. The arrival time priority degree of task $_{k_i}$ is given as follows:

$$AI_{k_i} = \frac{AS_{k_i}}{WS_{k_i}}. \quad (4)$$

The deadline priority degree of task $_{k_i}$ is as follows:

$$DI_{k_i} = \frac{DS_{k_i}}{WS_{k_i}}. \quad (5)$$

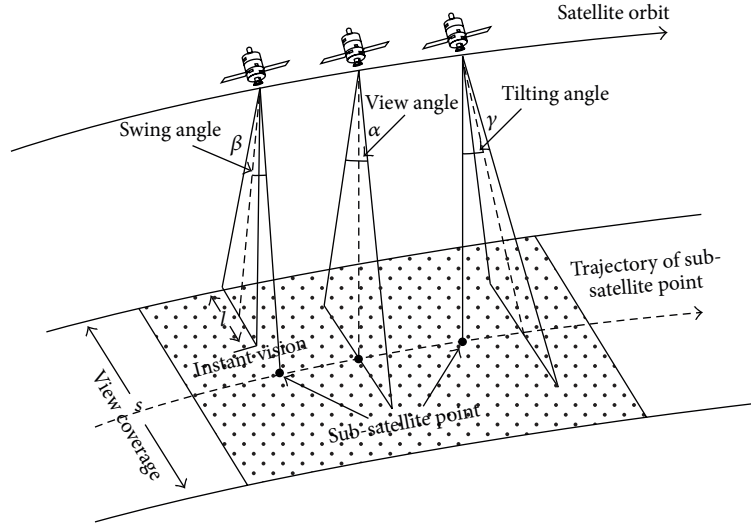


FIGURE 1: The observation field of EOS.

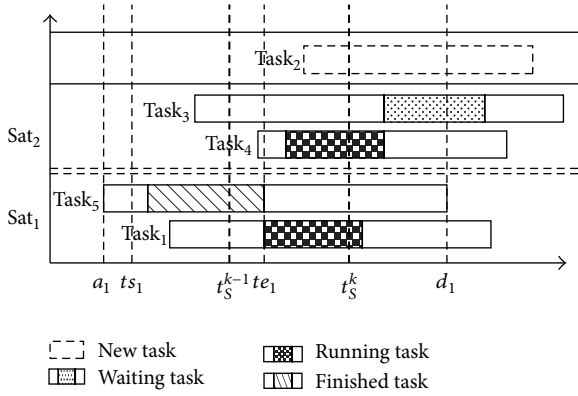


FIGURE 2: Task states based on current scheduling time.

The waiting time priority degree of task_{k_i} is as follows:

$$CI_{k_i} = \frac{CS_{k_i}}{WS_{k_i}}. \quad (6)$$

The main steps of the heuristic algorithm adopted in this paper are described as follows.

Step 1. The three priority degrees of each task in rolling-horizon are calculated.

Step 2. The tasks in the rolling-horizon are sorted by their different priority-degrees, and the ranking results can be obtained corresponding to the AIS, DIS, and WIS algorithms, respectively.

Step 3. The assignment strategy based on windows conflict index (WCI) is used to assign satellites and execution time for each task based on ranking results, and the scheduling scheme is generated after all tasks have been assigned.

In the aforementioned steps, WCI denotes the total impact on the unassigned tasks when a task is allocated to an available opportunity window. If task_i is appointed to the k_{th} opportunity window of sat_j, the WCI of task_i can be calculated as follows:

$$WCI_{i,j}^k = \sum_{task_{i'} \in \text{Waiting}_i, k'=1}^{q_{i',j}} \frac{g(tw_{i,j}^k, tw_{i',j}^{k'})}{we_{i',j}^{k'} - ws_{i',j}^{k'}} P_{i'}, \quad (7)$$

where Waiting_i is used to store tasks scheduled after task_i according to the ranking result and $g(tw_{i,j}^k, tw_{i',j}^{k'})$ can be calculated as follows:

$$g(tw_{i,j}^k, tw_{i',j}^{k'}) = \begin{cases} 0, & \text{if } tw_{i,j}^k \cap tw_{i',j}^{k'} = \emptyset, \\ \min(we_{i,j}^k, we_{i',j}^{k'}) - \max(ws_{i,j}^k, ws_{i',j}^{k'}), & \text{otherwise.} \end{cases} \quad (8)$$

Let $\text{Queue} = \{\text{task}_{b1}, \text{task}_{b2}, \dots, \text{task}_{bf}\}$ denote the task ranking result and $E\text{Queue}$ the finished task set delivered by Algorithm 1. The pseudocode of heuristic algorithm is described in Algorithm 2.

In Algorithm 2, the tasks in the rolling horizon are scheduled in order (see line 1), and the timing complexity is $O(m)$. The valid opportunity windows of task_i are checked to analyze the confliction with the running tasks (see lines 3~9); the timing complexity is $O(Tnm)$. It takes $O(Tn)$ to calculate the WCI of the valid opportunity windows (see lines 11~13). The timing complexity of other lines is $O(1)$, so the timing complexity of Algorithm 2 is $O(m)[O(Tnm) + O(Tn)] = O(Tnm^2)$. Furthermore, the timing complexity of Algorithm 1 is added to Algorithm 2; then the dynamic scheduling algorithm proposed in this paper has a timing complexity $O(P)[O(m) + O(Tnm^2)] = O(PTnm^2)$. Specially, the scheduling number and the reconnaissance activity

```

(1) for each timing instant  $t_s^k$  in set  $T_s$  do
(2)   RH  $\leftarrow$  NULL; Finished  $\leftarrow$  NULL; /* Initialization */
(3)   for each task  $task_i$  arrived before  $t_s^k$  do
(4)     if  $\exists x_{i,j}^k \leftarrow 0, tb_i \leftarrow$  NULL,  $te_i \leftarrow$  NULL then /* Delete the planning scheme of  $task_i$  */
(5)       Add  $task_i$  to set RH; /* RH is the rolling-horizon */
(6)     else if  $\sum_{j=1}^n \sum_{k=1}^{q_{i,j}} x_{i,j}^k = 0$  and  $a_i > t_s^{k-1}$  then /*  $task_i$  is a waiting task */
(7)       Add  $task_i$  to set RH;
(8)     end if
(9)   end for
(10)  Sort all task in set RH, and schedule each task by Heuristic
      Algorithm in order; /* The pseudocode of this algorithm is given in the following Algorithm 2 */
(11) Add  $task_i$  to set Finished;
(12) Update the scheduling decisions;
(13) end for

```

ALGORITHM 1: The pseudocode of RHO strategy.

```

(1) for each  $task_{b_i}$  in RH do
(2)   Validwindow  $\leftarrow$   $w_{b_i}$ ;  $ts_{b_i} \leftarrow$  NULL;  $te_{b_i} \leftarrow$  NULL;
(3)   for each  $tw_{b_i,j}^k$  in time window set Validwindow do
(4)     for each  $task_{b_i}$  in task set Finished do
(5)       if  $[w_{s_{i,j}^k}, we_{i,j}^k] \cap [ts_i, te_i] \neq \emptyset$  then
(6)         Delete  $tw_{i,j}^k$  from time window set Validwindow;
(7)       end if
(8)     end for
(9)   end for
(10)  Remove  $task_{b_i}$  from RH to Finished;
(11)  if time window set Validwindow  $\neq \emptyset$  then
(12)    for each  $tw_{i,j}^k$  in time window set Validwindow do
(13)      Calculate  $WCI_{i,j}^k$  of  $tw_{i,j}^k$ ;
(14)    end for
(15)    Assign  $task_{b_i}$  by WCI strategy, and calculate  $ts_i, te_i$ ;
(16)  end if
(17) end for

```

ALGORITHM 2: The pseudocode of heuristic algorithm.

period have a proportional relation if the rescheduling time interval is constant, then the timing complexity of dynamic scheduling algorithm is $O(T^2 m^2 n)$.

We incorporate RH strategy with AIS, DIS, and WIS to yield three new algorithms named RH-AIS, RH-DIS, and RH-WIS, respectively. Meanwhile, AIS, DIS, and WIS can also be used to solve the dynamic scheduling problem separately; that is, only new tasks are scheduled in each scheduling by those heuristic algorithms. The six algorithms mentioned before are compared in the following experiment to evaluate the efficiency of the RH strategy.

4. Evaluation

The proposed algorithms are implemented by Matlab2007 on a laptop with Pentium IV 3.06 GHz CPU, 2 GB memory, and Windows XP operating system. The experimental scenarios are generated randomly for there has been no benchmark in

the field of satellite scheduling yield. The operating points of simulated experiment are given as follows [22–24].

- (1) The reconnaissance activity period is from March 21, 2010, to March 22, 2010, and the scheduling period is two hours; that is, $T_R = \{0 \text{ h}, 2 \text{ h}, \dots, 24 \text{ h}\}$.
- (2) The imaging tasks are generated in the area with a longitude $0^\circ \sim 150^\circ$ and latitude $-30^\circ \sim 60^\circ$ randomly. The task quantity varied from 100 to 400. The arrival time gap between two adjacent tasks is subject to the negative exponential distribution, with a density of 0.1. Set the execution value of task from 1 to 10, the required continuous time 3~5 minute, and the occupied storage 2~4 G, and the deadline is a random variable generated between the arrival time of the task and the ending time of the observation activity, which abides by the uniform distribution.

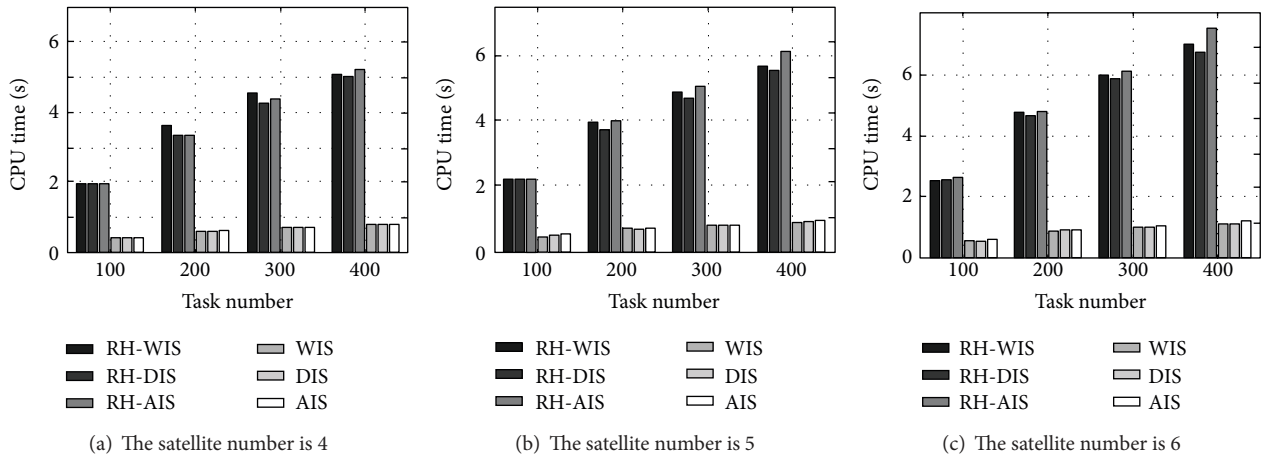


FIGURE 3: Search time of algorithm in different satellite quantities.

- (3) The satellite quantity varied from 4 to 6, the memory storage is 240 G, the field angle is 3°, the maximum sway angle is 35°, the maximum tilting angle is 40°, and the maximum number of position transfers within a single orbit is no more than 5.

For the convenience of description, the dynamic scheduling problem of n satellites and m tasks is recorded as $m \times n$. The twelve experiments are designed in different problem scales, and the performance metrics include the task benefits and guarantee ration. The scheduling results of various algorithms are displayed in Table 1.

From Table 1, we can observe that the RH strategy can effectively improve the performance of heuristic algorithms. The task benefits of the three heuristic algorithms embedding RH strategy increase by 16.19%, 15.82%, and 8.29%, respectively. This is because the algorithms without embedded RH strategy only operate the new tasks, while the embedded one can adjust the scheduling scheme of waiting tasks for executing the new tasks which can not be executed previously; hence a better scheduling solution can be achieved. Among the rolling-horizon scheduling algorithms (RH-AIS, RH-DIS, and RH-WIS), the task benefits obtained by RH-WIS are slightly higher than those of RH-DIS and RH-AIS while the satellites have adequate observation capacity to execute imaging tasks (for the problem scales 100×4 , 100×5 , and 100×6), and the resolution of RH-DIS is better than that of the other algorithms while the satellites have inadequate observation capacity. This is because both RH-DIS and RH-WIS have considered the deadline feature of tasks in scheduling scheme; it is the precondition to execute the valuable task. A large amount of high-timeliness tasks cannot be executed in their deadlines if the satellites have inadequate observation capacity, then RH-DIS gives priority to the tasks with earlier deadlines, so that the task guarantee ration is increased, and higher task benefits can be obtained. Most tasks can be effectively executed if the satellites have adequate observation capacity, then RH-AIS schedules tasks based on the deadline distribution but also considers the impact of executing current task on follow-up tasks, so that higher task

benefits can be obtained on the aspect of overall planning effects.

From Figure 3, the CPU time of six algorithms is compared in different satellite numbers. The timing complexity of scheduling algorithms is aggravated after embedding the RH strategy; thus the generation speed of the planning scheme is reduced. In the scenarios with three different satellite numbers, the average CPU times of algorithms incorporating RH strategy are increased 3.76, 3.73, and 3.74 times, respectively. The increased CPU time of rolling-horizon scheduling does not have special change with raising satellite number, which means the impact caused by RH strategy on the timing complexity of scheduling algorithm is weakly related to the satellite number, and this is consistent with the previous conclusion in the timing complexity analysis of Algorithm 2. Although the RH strategy will delay the generation of planning scheme, it also satisfies the timelessness of dynamic scheduling because the longest CPU time of the rolling-horizon scheduling is only 7.5 s, which is a low level.

Set the scheduling period from one to twelve hours in order to analyze the impact of scheduling time interval on the overall performance of planning algorithm. The six algorithms are tested in different problem scales and shown in Figure 4.

From Figure 4, the task benefits are reduced overall as the scheduling period extends gradually. This is because the extensive scheduling period will increase the rejected tasks for their deadline can not be satisfied; hence the low task benefits are obtained for satellites unable to execute the tasks in time. The heuristic algorithms which have not been embedded into RH strategy (including RH-AIS, RH-DIS, and RH-WIS) still have high task benefits in different scheduling period. Let the ratio between the task number n and the satellite number m as the approximate workload of the satellite resources; then a scheduling period between 1~8 h does not have a significant impact on the task benefits if the satellite resource has a light workload. However, the task benefits will be sharply reduced if the scheduling period is extended to 8~12 h. There are high task benefits if we maintain the scheduling period within 1~6 h to adapt the increased

TABLE I: Simulation results in different scenarios.

Problem scale	RH-CIS			RH-DIS			RH-AIS			CIS			DIS			AIS		
	Task benefits	Guarantee ration	Task benefits	Guarantee ration	Task benefits	Guarantee ration	Task benefits	Guarantee ration	Task benefits	Guarantee ration	Task benefits	Guarantee ration	Task benefits	Guarantee ration	Task benefits	Guarantee ration	Task benefits	Guarantee ration
100 × 4	396	0.672	401	0.683	397	0.674	382	0.642	397	0.677	396	0.673	396	0.677	396	0.673	396	0.673
100 × 5	412	0.705	417	0.716	416	0.713	401	0.681	416	0.713	409	0.698	416	0.713	409	0.698	416	0.713
100 × 6	425	0.733	430	0.745	424	0.729	408	0.696	426	0.726	424	0.729	426	0.726	424	0.729	424	0.729
200 × 4	679	0.622	672	0.613	633	0.574	559	0.497	559	0.497	554	0.492	559	0.497	554	0.492	554	0.492
200 × 5	746	0.691	740	0.682	699	0.638	611	0.553	611	0.553	610	0.550	611	0.553	610	0.550	610	0.550
200 × 6	804	0.741	813	0.754	763	0.707	688	0.625	688	0.625	693	0.637	688	0.625	693	0.637	693	0.637
300 × 4	808	0.492	813	0.506	725	0.442	665	0.409	663	0.405	658	0.392	663	0.405	658	0.392	658	0.392
300 × 5	890	0.551	900	0.562	812	0.506	745	0.458	744	0.453	736	0.450	744	0.453	736	0.450	736	0.450
300 × 6	1027	0.649	1021	0.642	933	0.580	873	0.541	877	0.549	860	0.531	877	0.549	860	0.531	860	0.531
400 × 4	897	0.423	895	0.413	801	0.366	729	0.339	722	0.332	717	0.326	722	0.332	717	0.326	717	0.326
400 × 5	992	0.468	990	0.461	876	0.403	815	0.377	802	0.370	800	0.364	802	0.370	800	0.364	800	0.364
400 × 6	1152	0.549	1143	0.541	1025	0.487	959	0.448	953	0.441	946	0.435	953	0.441	946	0.435	946	0.435

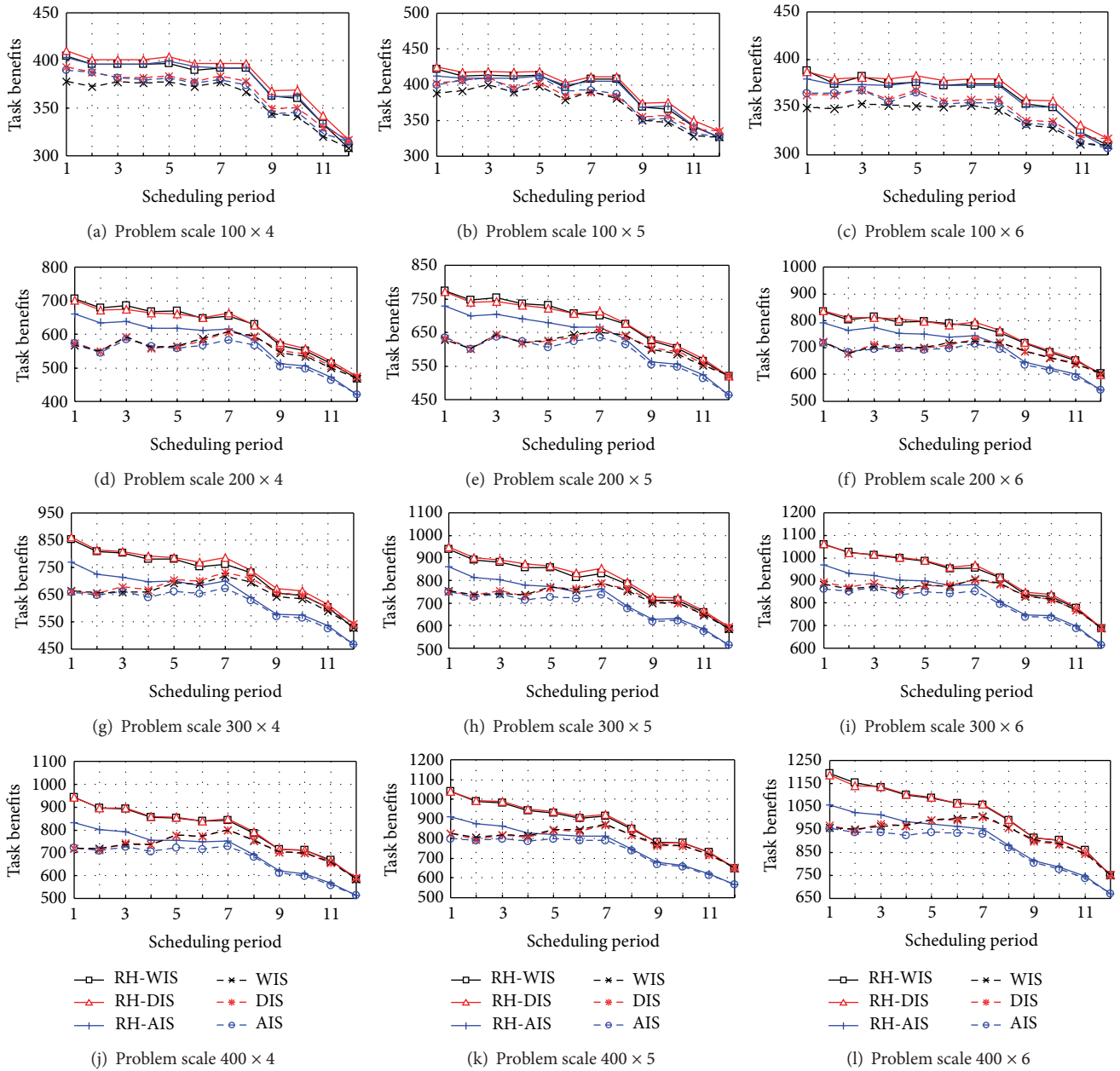


FIGURE 4: Performance of algorithms in different problem scales.

workload of the satellite resource. It is also appropriate to maintain the scheduling period within 1~3 h in accordance with the heavier workload. Furthermore, it should be ensured the scheduling period is no more than 2 h while the satellite resource has the heaviest workload. So far we conclude that the scheduling time is a key factor to affect the task benefits. In addition, the task benefits do not present monotonic change in accompany with the scheduling period as shown in Figure 4. For example, Figure 4(a) presents that the task benefits gradually increase with the extension of scheduling period within 1~4 h and fluctuate within 4~8 h but begin to reduce within 8~12 h. This is because there is a conflict

between the performance scheduling requirement of the planning system and the timeliness scheduling requirement of the tasks. The scheduling system only acquires the local task information if the scheduling period is short and yields inferior scheme which can not handle the following tasks with high timelessness. On the contrary, a large amount of high-timeliness tasks may not be finished before their deadline if the scheduling period is long, and the task benefits are reduced due to a low guarantee ration. Therefore, the scheduling period should be reasonably selected based on the satellite quantity and task density in actual application, so as to ensure the performance of scheduling system.

5. Conclusions and Future Work

This paper has studied the dynamic scheduling problem of EOSs. An integer programming model has been constructed by considering the independent arrival time and deadline of the imaging tasks. A dynamic scheduling algorithm based on the RH strategy which can be combined with multiple heuristic algorithms proposed in many researches and its timing complexity has been analyzed.

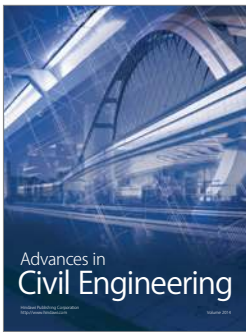
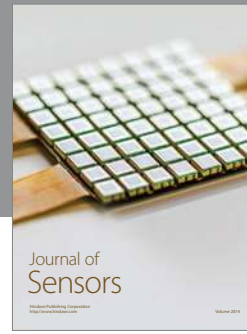
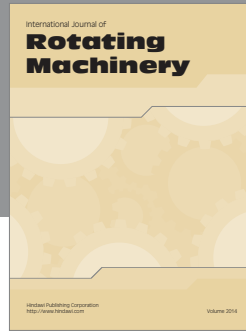
The scheduling algorithms adapted to RH strategy can effectively adjust to the planning scheme based on the satellite workload to execute the emergency tasks. The effectiveness of this strategy has been verified by comparing the scheduling results of six algorithms in the experiment. It is worth to restate that the RH strategy might cause a high time consumption to yield an optimized scheme if a large amount of tasks are involved in the scheduling system. This problem can be solved by limiting the size of the rolling horizon.

Also for our future work, we plan to research the detection method of scheduling time, which is a significant factor to impact the performance of scheduling algorithm and system. With the method in place, we will extend our algorithm to cooperative scheduling of EOSs; we will consider Qos requirements in our RH-WIS; we are going to combine the dynamic resources management into our scheme.

References

- [1] 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.
- [2] P. Wang and Y. Tan, "A heuristic method for selecting and scheduling observations of satellites with limited agility," in *Proceedings of the 7th World Congress on Intelligent Control and Automation (WCICA'08)*, pp. 5292–5297, June 2008.
- [3] T. Y. Mao, Z. Q. Xu, R. Hou et al., "Efficient satellite scheduling based on improved vector evaluated genetic algorithm," *Journal of Networks*, vol. 7, no. 3, pp. 517–523, 2012.
- [4] 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.
- [5] S. M. Han, S. W. Beak, K. R. Cho, D. W. Lee, and H. D. Kim, "Satellite mission scheduling using genetic algorithm," in *Proceedings of the International Conference on Instrumentation, Control and Information Technology (SICE '08)*, pp. 1226–1230, August 2008.
- [6] H. Wang, M. Xu, R. Wang, and Y. Li, "Scheduling earth observing satellites with hybrid ant colony optimization algorithm," in *Proceedings of the International Conference on Artificial Intelligence and Computational Intelligence (AICI '09)*, vol. 2, pp. 245–249, November 2009.
- [7] L. Jian and W. Cheng, "Resource planning and scheduling of payload for satellite with genetic particles swarm optimization," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '08)*, pp. 199–203, June 2008.
- [8] C. He, X. M. Zhu, and D. S. Qiu, "Cooperative scheduling method of multi-satellites for imaging reconnaissance in emergency condition," *Systems Engineering and Electronics*, vol. 34, no. 4, pp. 721–731, 2012.
- [9] S. Baolin, W. Wenxiang, and Q. Qianqing, "Satellites scheduling algorithm based on dynamic constraint satisfaction problem," in *Proceedings of the International Conference on Computer Science and Software Engineering (CSSE '08)*, vol. 4, pp. 167–170, December 2008.
- [10] K. J. Zhu, J. F. Li, and H. X. Baoyin, "Satellite scheduling considering maximum observation coverage time and minimum orbital transfer fuel cost," *Acta Astronautica*, vol. 66, no. 1-2, pp. 220–229, 2010.
- [11] J. M. Wang, J. F. Li, and Y. J. 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)*, vol. 1, pp. 9–14, August 2007.
- [12] 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.
- [13] D. T. Chi and Y. T. Su, "On a satellite coverage problem," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 31, no. 3, pp. 891–896, 1995.
- [14] S. K. Chronopoulos, C. T. Angelis, A. Koumasis, and P. Drakou, "Satellite coverage analysis for the investigation of real-time communication in selected areas," *WSEAS Transactions on Communications*, vol. 5, no. 10, pp. 1965–1972, 2006.
- [15] G. Verfaillie, C. Pralet, and M. Lemaître, "Constraint-based modeling of discrete event dynamic systems," *Journal of Intelligent Manufacturing*, vol. 21, no. 1, pp. 31–47, 2010.
- [16] C. He, X. M. Zhu, H. Guo et al., "Rolling-horizon scheduling for energy constrained distributed real-time embedded systems," *Journal of Systems and Software*, vol. 85, no. 4, pp. 780–794, 2012.
- [17] 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.
- [18] L. Tang, S. Jiang, and J. Liu, "Rolling horizon approach for dynamic parallel machine scheduling problem with release times," *Industrial and Engineering Chemistry Research*, vol. 49, no. 1, pp. 381–389, 2010.
- [19] U. Schmid and J. Bliederger, "Some investigations on FCFS scheduling in hard real time applications," *Journal of Computer and System Sciences*, vol. 45, no. 3, pp. 493–512, 1992.
- [20] M. Bertogna and S. Baruah, "Tests for global EDF schedulability analysis," *Journal of Systems Architecture*, vol. 57, no. 5, pp. 487–497, 2011.
- [21] F. Zhang and A. Burns, "Schedulability analysis for real-time systems with EDF scheduling," *IEEE Transactions on Computers*, vol. 58, no. 9, pp. 1250–1258, 2009.
- [22] Y. Chen, M. Q. Zhou, and H. Zou, "Multi-satellite observation scheduling algorithm based on hybrid genetic particle swarm optimization," *Lecture Notes in Electrical Engineering*, vol. 136, pp. 441–448, 2012.
- [23] C. He, D. S. Qiu, and J. Liu, "Cooperative scheduling of imaging observation tasks for high-altitude airships based on propagation algorithm," *The Scientific World Journal*, vol. 2012, Article ID 548250, 13 pages, 2012.

- [24] W. C. Lin and D. Y. Liao, "A tabu search algorithm for satellite imaging scheduling," in *Proceedings of IEEE International Conference on Systems, Man and Cybernetics (SMC '04)*, vol. 2, pp. 1601-1606, October 2004.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

