Journal of Systems Engineering and Electronics
Vol. 23, No. 5, October 2012, pp.723–733

Multi-satellite observation integrated scheduling method oriented to emergency tasks and common tasks

Guohua Wu, Manhao Ma, Jianghan Zhu, and Dishan Qiu*

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

Abstract: Satellite observation scheduling plays a significant role in improving the efficiency of satellite observation systems. Although many scheduling algorithms have been proposed, emergency tasks, characterized as importance and urgency (e.g., observation tasks orienting to the earthquake area and military conflict area), have not been taken into account yet. Therefore, it is crucial to investigate the satellite integrated scheduling methods, which focus on meeting the requirements of emergency tasks while maximizing the profit of common tasks. Firstly, a pretreatment approach is proposed, which eliminates conflicts among emergency tasks and allocates all tasks with a potential time-window to related orbits of satellites. Secondly, a mathematical model and an acyclic directed graph model are constructed. Thirdly, a hybrid ant colony optimization method mixed with iteration local search (ACO-ILS) is established to solve the problem. Moreover, to guarantee all solutions satisfying the emergency task requirement constraints, a constraint repair method is presented. Extensive experimental simulations show that the proposed integrated scheduling method is superior to two-phased scheduling methods, the performance of ACO-ILS is greatly improved in both evolution speed and solution quality by iteration local search, and ACO-ILS outperforms both genetic algorithm and simulated annealing algorithm.

Keywords: satellite scheduling, emergency task, ant colony optimization (ACO), iteration local search (ILS), acyclic directed graph model.

DOI: 10.1109/JSEE.2012.00089

1. Introduction

An earth observing satellite (EOS) orbits the globe and acquires images by observation sensors, which has become the important means for earth reconnaissance and resources researches. Currently, many countries over the world tend to increase investment to develop EOS and corresponding techniques. Although the number of EOS is continuously increasing, it is still insufficient to satisfy the requirements of various users. Therefore, it is necessary to develop effective satellite observation scheduling methods to make full use of the limited EOS resources so as to better satisfy users' demands. Satellites observation scheduling is to reasonably assign time-windows and satellite resources to earth observation tasks on the precondition of satisfying complex constraints, which has been proved to be NP-complete [1].

Regarding to this complex issue, the early studies concentrated on single satellite observation scheduling. Potter and Gasch adopted an improved greed algorithm to mission planning and scene acquisition scheduling for the Landsat 7 satellite [2]. Bensana et al. viewed the single satellite scheduling problem as a value constraint-satisfaction problem, and used exact methods like depth first branch and bound or Russian dolls search and approximate methods like greedy search or tabu search (TS) to solve the scheduling problem of the SPOT5 satellite [3]. Through experimental comparisons, they drew the conclusion that the exact algorithm was only suitable for tackling small scale scheduling problems, and approximate methods were more effective in dealing with the large scale problems. Vasquez and Hao formulated the single satellite observation scheduling problem into a generalized version of the well-known knapsack model. Furthermore, they proposed a TS algorithm to solve the built problem [4]. In another paper, Vasquez and Hao presented a Partition-based approach (UPPB) to address the single satellite observation scheduling problem. They divided the problem into multiple sub-problems, and each sub-problem was solved by enumeration algorithms, respectively [5]. Harrison et al. built the imaging model of the optical or radar based earth observing satellite, and proposed a enumerate search method to solve the problem in small scale [6]. Lin et al. adopted the mathematical programming method to acquire a near-optimal schedule of the problem [7]. Gabrel

Manuscript received May 26, 2011.
*Corresponding author.
This work was supported by the National Natural Science Foundation of China (61104180) and the National Basic Research Program of China (973 Program) (97361361).

and Murat presented two upper bound procedures [8]. The first one was based on the graph theory and the other on a column generation technique on a vertex-path formulation, and both of them were tested on single satellite scheduling problems. Gabrel and Vanderpooten proposed an acyclic graph model to formulate the scheduling problem of SPOT5, then generated the efficient paths and selected a satisfactory path by a multiple criteria interactive procedure [9].

Literature above basically focused on single satellite observation scheduling, while multi-satellite observation scheduling problems, which are more complicated and changeable, have intrigued an increasing number of researches' interests. Wang et al. proposed a multi-objective EOS imaging scheduling method based on the strength Pareto evolutionary algorithm 2 (SPEA2) to deal with the multi-satellite observation scheduling problem [10]. Mansour et al. developed a genetic algorithm (GA) to solve the SPOT5 scheduling problem using a new genome representation for maximizing not only a single objective as profits but also a multi-criteria objective that included the number of acquired photographs [11]. Bianchessi et al. employed a TS heuristic for the problem of selecting and scheduling the observation requests to be satisfied for multiple satellites under diverse constraints, and adopted an upper bounding procedure based on column generation to evaluate the quality of the solutions [12]. Frank et al. adopted a constraint-based interval (CBI) framework to represent the resources of EOS and proposed a heuristic for guiding this search procedure based on a general notion of contention for resources, but they did not consider the conflicting requests and considered the resources are independent [13]. Lemaitre et al. investigated the problem of scheduling the set of photographs for agile earth observation satellites, and found that constraint programming was more flexible while local search could find better solutions [14]. Qiu et al. proposed the first finish first schedule with discard task moving back (FFFS-DTMB) and accommodate discard task predicting coexistence with discard task moving back (ADTPC-DTMB) algorithms to solve multi-satellite observation scheduling problems [15].

From the previous survey, we can draw some conclusions. Firstly, early studies mainly emphasized on single satellite observation scheduling problems, while the multi-satellite observation scheduling problem is drawing more researchers' attention. Secondly, exact algorithms can obtain optimal solutions for relative simple satellite observation scheduling problems, while approximate algorithms are more efficient and adaptable in tackling large size one. Thirdly, nearly all studies aim at common tasks, whereas emergency tasks, characterized by importance and urgency, exist in real-world applications of satellites frequently. Because of different demands between common tasks and emergency tasks, these two kinds of tasks should be treated differently in the scheduling process. Therefore, it is necessary to develop an effective scheduling method that can cope with common tasks and emergency tasks synthetically. Finally, many heuristic algorithms (e.g., local search, simulated annealing (SA), TS, GA, etc.) have been employed to solve satellite observation scheduling problems, however, ant colony optimization (ACO), as a well-performed algorithm in solving complex combinatorial optimization problems, has not been applied to this field yet.

When scheduling the satellite observation aiming at common tasks, the common tasks are given priority values in advance usually, and then the objective of satellite observation scheduling is to make the sum priorities of the tasks arranged for observation maximally. In fact, emergency observation tasks (e.g., observation tasks orienting to the earthquake area, forest fire disaster area, and military conflict area, etc.) exist widely and should not be neglected. Owing to their importance and urgency, emergency tasks should get preferential treatment. Hence the execution of emergency tasks should be set as a hard constraint instead of be controlled by the priority which may cause the emergency task to fail in competing for observation though they have potential time-windows. That is to say, as long as an emergency task has available time-window and dose not conflict with other emergency tasks, this task must be arranged in the final solution. A straightforward way with two phases to resolve the scheduling problem including common tasks and emergency tasks is to assign satellites and time-window resources to each emergency task according to a heuristic rule firstly, then rationally to schedule the common tasks based on the remaining observation resources. Although this two-phased method can guarantee emergency tasks being accommodated, the heuristic assignment strategy in the first stage may destroy the optimization of solutions. As a result, an integrated scheduling method instead of the two-phased method is needed.

The major contributions of this paper are summarized as follows.

(i) For the first time, an integrated multi-satellite observation scheduling framework is proposed, aiming at both common tasks and emergency tasks.

(ii) A pretreatment method is put forward to match tasks requirements to resources observation capability, distribute possible tasks to each orbit of satellites, and eliminate conflicts among emergency tasks.

(iii) An acyclic directed graph model is established to describe multi-satellite observation scheduling problems according to the possible observation order in each orbit of satellites.

(iv) A novel hybrid ACO mixed with iteration local search (ACO-ILS) is presented to solve the problem on the basis of the graph model. Besides, a constraint repair strategy is developed to ensure that each solution obtained by ants satisfies emergency tasks requirement constraints.

The remainder of the paper is organized as follows. In the next section the corresponding problem is described and the solving framework is presented. Section 3 presents the pretreatment procedure for observation tasks and observation resources. Section 4 constructs a mathematical model and an acyclic directed graph model for the problem. Section 5 proposes ACO-ILS and expounds its detailed factors, respectively. Section 6 conducts extensive experimental simulations and performance analyses. Section 7 provides the conclusions and direction of the future work.

2. Problem description and solving framework

Observing satellites, orbiting the earth, can observe targets around the sub-satellite track through adjusting the slew angle of their sensors. As shown in Fig. 1, satellite's available observation tasks sequence may be $1 \to 3 \to 4 \to 6 \to 7 \to 8$.

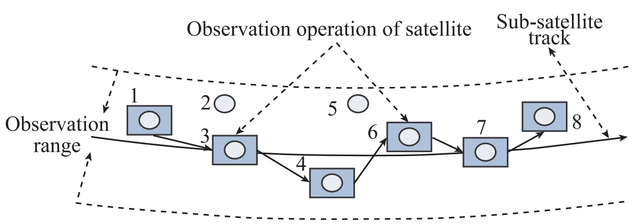


Fig. 1 Observation process of satellite

From Fig. 1, we can deduce that the purpose of satellite observation scheduling is to arrange the corresponding task sequence for each orbit of satellites, which attempts to make the sum of the priorities of all tasks in observation sequence maximal while satisfying complex constraints.

The time-window between each task and each orbit of satellites can be computed before scheduling. Hence the different orbits of all satellites can be viewed as a similar kind of resources with the respective observation capability. To describe the problem more clearly, we fairly organize and code all orbits together without considering which satellite these orbits belong to. As a result, the multi-satellite scheduling problem could be transformed into a multi-orbit scheduling problem. Let $O = \{o_i | i = 1, 2, \dots, |O|\}$ be the orbit set, where $|O|$ denotes the total number of orbits.

Fig. 2 depicts the problem solving framework developed in this paper. Firstly, the pretreatment operation for observation tasks and observation resources is conducted, which can effectively decrease the complexity of the initial problem. Secondly, an acyclic directed graph for multi-satellite observation scheduling problems will be constructed based on the pretreatment results. Thirdly, a hybrid ACO-ILS and constraint repair will be developed to find qualified solutions for the scheduling problem. The following sections will elaborate the key techniques of the solving framework.

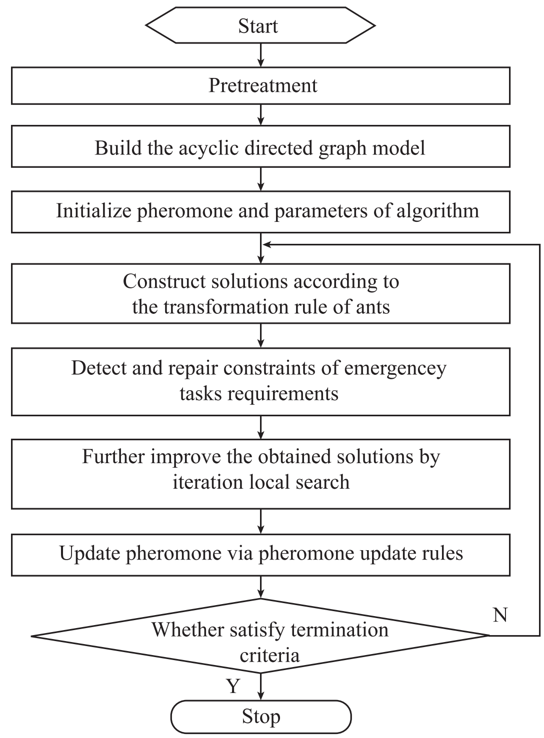


Fig. 2 Problem solving framework

3. Pretreatment of tasks and resources

Pretreatment firstly computes the time-window and resolution for each observation task according to satellite orbit parameters, sensor parameters and observation target position, then according to the computation results, matches tasks to related observation resources (orbits). In this process, the tasks without available time-windows or violating corresponding constraints, such as resolution power constraints, will be discarded in advance, while the other tasks will be allocated to related orbits. It should be noticed that the same task may be allocated to multiple orbits, if any of these orbits could finish this task. Which orbit will execute the task finally depends on the scheduling algorithm described below.

In the pretreatment phase, the conflicts among emergency tasks allocated to the same orbit will be gotten rid of. If there are conflicts among emergency tasks, it means some emergency tasks cannot be observed. Therefore, at this stage, we should decide which emergency tasks are to be remained for the latter scheduling, and which emergency tasks are to be discarded in advance. Besides, some special emergency tasks might be assigned to the given orbits according to decision-maker's preference, which enables planners manually adjusting the scheduling process to enhance the scheduling flexibility.

To eliminate the conflicts between two emergency tasks, we establish a conflict task set for each emergency tasks. The conflict task set of one emergency task consists of the tasks conflicting with this emergency task. The conflict priority of an emergency task is defined as the sum priorities of the tasks included in its conflict task set. When two emergency tasks conflict with each other, the one with higher conflict priority will be discarded accordingly. For example, assume that there are conflicts between emergency tasks t_i and t_k, and their conflict priorities are pr_i and pr_k, respectively. If $pr_i \geqslant pr_k$, it means the accomplishment of t_i will generate more negative influence on other tasks, and t_i would be deleted on the pretreatment phase. Therefore, the conflicts between t_i and t_k are eliminated. Let $T = \{t_i | i = 1, 2, \dots, |T|\}$ be the tasks set requiring observation, where $|T|$ denotes the number of tasks included in T. Let TA_j be the task set and TE_j be the emergency task set, where TA_j includes the tasks distributed to orbit, and TE_j consists of the emergency tasks included in TA_j. The pseudocode of the pretreatment algorithm is shown as follows:

```
set ∀ TA_j = ∅, ∀ TE_j = ∅;
for j = 1 → |O| do
  for i = 1 → |T| do
    compute the time-window between o_j and t_i;
    if the time-window exists between the o_j and t_i do
      add t_i to TA_j;
      if t_i is an emergency task do
        add t_i to TE_j;
      end if
    end if
  end for
end for
for j = 1 → |O| do
  sort tasks in TE_j ascendly according to the beginning
  time-window;
  for ∀t_i, t_k ∈ TE_j, i ≠ k do
    if conflicts exist between t_i and t_k do
      delete the task with more conflict priorities from
      TE_j and TA_j;
    end if
  end for
end for
```

4. Multi-satellite observation scheduling model

Complicated constraints need to be satisfied in the scheduling process, and detailed descriptions of these constraints were given by Verfaillie et al. [16] and Globus et al. [17]. Let $tw_i = \{tws_i, twe_i\}$ be the time-window of task t_i, where tws_i is the start time, and twe_i is the end time. Assume that p_i is the priority of task t_i, e_i is the energy consumption of task t_i, e_{ik} is the energy consumption by sensor's slewing switch from t_i to t_k, c_i is the memory consumption of task t_i, v_j is sensor's slewing velocity orbit o_j, sta_j is the stabilization time of sensor slewing, E_j is maximal energy can be consumed in orbit o_j, C_j is maximal memory can be consumed in orbit o_j. Task set TO_j contains the tasks arranged for observation in orbit o_j.

The scheduling objective is to maximize the sum priorities of finished tasks, which is defined as

$$\max\left(\sum_{i=1}^{|T|} p_i x_i\right), \quad x_i = \begin{cases} 1, & \exists\, TO_j,\ t_i \in TO_j \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

where x_i is a mark variable identifying whether t_i has been arranged in one solution.

Uniqueness constraint: one task needs only one observation in one orbit.

$$\forall\, TO_i,\ TO_j\ (i \neq j),\ TO_i \cap TO_j = \varnothing. \tag{2}$$

Sensor slewing switch time constraint: if the observation operation transforms from one task to another consecutively, there must exist enough time for adjusting and stabilizing the sensor's slew angel.

$$\forall\, o_j\ (j = 1, 2, \dots, |O|),\ \ \forall\, t_i, t_k \in TO_j$$
$$\frac{|\theta_i - \theta_k|}{v_j} + sta_j \leqslant tws_k - twe_i \tag{3}$$

where t_k is observed after t_i in orbit o_j.

Energy constraint: the maximal energy that satellite could consume in an orbit is limited.

$$\forall\, o_j\ (j = 1, 2, \dots, |O|)$$
$$\sum_{t_i \in TO_j} e_i + \sum_{t_i, t_k \in TO_j} e_{ik} y_{ik} \leqslant E_j \tag{4}$$

where if t_k is observed after t_i in orbit o_j, $y_{ik} = 1$, otherwise, $y_{ik} = 0$.

Memory capacity constraint: the maximal memory that satellite could consume in an orbit is limited.

$$\forall\, o_j\ (j = 1, 2, \dots, |O|),\quad \sum_{t_i \in TO_j} c_i \leqslant C_j. \tag{5}$$

Emergency task requirement constraint: if an emergency task has one or more satisfactory time-windows after the pretreatment procedure, this emergency task must be arranged into the final schedule.

$$\forall\, t_i,\ \exists\, SE_j,\ t_i \in SE_j,\ x_i = 1. \tag{6}$$

If a task has a time-window in orbit o_j, we view this task as a possible observation task of orbit o_j. According to the possible observation tasks in each orbit, we construct an acyclic directed graph model, which is shown in Fig. 3. G_j is the sub acyclic directed graph relating to orbit o_j, and all G_j together describe the whole scheduling problem. G_j depicts the relations among observation tasks allocated to orbit o_j. In G_j, each vertex is corresponding to a possible observation task in TA_j, and directed edges reveal the possible order of different observation tasks. As illustrated in G_1, after having observed t_a, if a satellite still has enough time to adjust the sensor's slew angle to observe t_b, then a directed edge is added from t_a to t_b. That is to say, in orbit o_j, t_b is a possible subsequent task of t_a.

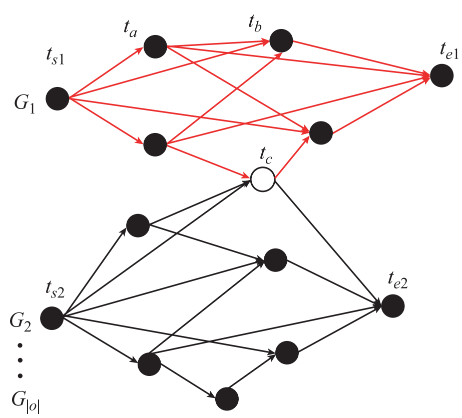


Fig. 3 Acyclic directed graph model of multi-satellites observation scheduling

To analyze the problem more conveniently, both a dummy start vertex t_{sj} and a dummy end vertex t_{ej} are added to each G_j. Any dummy vertex does not consume any resource. It should be noted that the same vertex may belong to multiple different sub graphs, for the reason that the similar task could be observed in multiple different orbits. For example, in Fig. 3, task t_c can be observed by either o_1 or o_2, so t_c will be distributed to both G_1 and G_2 simultaneously.

5. Hybrid ACO

ACO, proposed by professor Dorigo initially [18], now is widely applied to diverse fields, such as traveling salesman problems [19], vehicle routing problems [20], and flow shop scheduling problems [21]. ACO is a meta-heuristic algorithm in which a colony of artificial ants cooperates in finding good solutions to complicated discrete optimization problems [22].

Based on the acyclic directed graph model, the problem solved process could be transformed into searching a feasible route of high quality in the graph for each orbit. This route searching process is naturally similar to route optimization process of ants. Consequently, we develop the ACO-ILS to resolve multi-satellite observation scheduling problems. The algorithm framework of ACO-ILS is described as follows.

```
set parameters, initialize pheromone values;
while (termination condition not met) do
  for i = 1 → AntSize do
    construct feasible soulution for each ant;
    apply iteration local search to each acquired
    solution;
  end for
  update pheromone;
end while
```

5.1 Construction of feasible solution

Vertex transference rules guide the solution construction process of ants. Assume that $Allow_{ji}^{m}$ is a subsequent candidate task set of t_i, when the mth ant prepares to transfer from task t_i to a next subsequent task in orbit o_j.

$Allow_{ji}^{m}$ contains the feasible subsequent tasks of t_i. Any task t_l to be included in $Allow_{ji}^{m}$ should satisfy the following three conditions: (i) there exists a directed edge from t_i to t_l, (ii) t_l has not been arranged into the ant's solution yet, i.e., $x_l \neq 1$ so far, (iii) if t_l is inserted into the current solution, the solution still meets energy constraints and memory capacity constraints.

The transference rule that the mth ant transfers from task t_i to the next feasible task is depicted as: if $Allow_{ji}^{m} \cap TE_j \neq \varnothing$ (there exist emergency tasks in $Allow_{ji}^{m}$), the ant will choose the earliest emergency task of $Allow_{ji}^{m}$ as the next vertex (observation task) of t_i, otherwise (i.e., $Allow_{ji}^{m} \cap TE_j = \varnothing$), the ant selects a task $t_k \in Allow_{ji}^{m}$ as the next observation task according to the following formula:

$$\Pr{}^{m}(o_j, t_i, t_k) = \frac{[\tau_{jik}]^{\alpha}[\eta_{jik}]^{\beta}}{\sum\limits_{l \in Allow_{ji}^{m}} [\tau_{jil}]^{\alpha}[\eta_{jil}]^{\beta}}, k \in Allow_{ji}^{m} \tag{7}$$

where τ_{jik} is the pheromone value for transferring from t_i to its subsequence task t_k in orbit o_j. Ants cooperate with each other through the accumulation and evaporation of pheromone, which can be represented by the weight of each edge. At the beginning, the initial pheromone value of each edge is set as τ_0. It should be noticed that the edges directed to emergency tasks have no pheromone because the execution of emergency tasks is not controlled by pheromone, as introduced in the transference rule above. $[\eta_{jik}]$ represents the heuristic information. α, β denote the importance degrees of pheromone and heuristic information, respectively.

To make effective use of various heuristic information of the scheduling problem, $[\eta_{jik}]$ is defined as follows:

$$\eta_{jik} = \frac{\lambda p_k}{\omega(tws_k - twe_i) + \upsilon \Delta\phi_{ik}} \quad (8)$$

where p_k is the priority of t_k, $(tws_k - twe_i)$ denotes the time span between t_i and t_k, and $\Delta\phi_{ik}$ denotes the necessary angle range of sensor slewing from t_i to t_k.

$\lambda, \omega, \upsilon$ represent the weight of these three kinds of heuristic information, respectively. The definition of η_{jik} implies that the common task with higher priority, shorter time span from the current task, and the small slewing angle from the current task are prone to be observed afterwards.

The procedure that the $k\mathrm{th}$ ant constructs feasible solution is depicted as the following steps.

Step 1 Judge whether orbit set O is empty. If $O = \varnothing$, the algorithm terminates. Otherwise, an orbit o_j is randomly selected from O and it is started to search the feasible route in sub graph G_j, and let $O = O - o_j$. Then, go to Step 2.

Step 2 Search route along with the directed edge in graph G_j according to ant's transference rules repeatedly, until no feasible subsequent tasks exist (i.e., for current task t_i, $Allow_{ji}^m = \varnothing$). Go to Step 1.

The pseudocode of the feasible solution construction algorithm is described as follows:

while $O \neq \varnothing$ do
  randomly select an obit $o_j \in O$;
  start to search route in graph G_j;
  add t_{sj} to TO_j;
  $t_i \leftarrow t_{sj}$;
  construct $Allow_{ji}^m$;
  while $Allow_{ji}^m \neq \varnothing$ do
    if $Allow_{ji}^m \cap TE_j \neq \varnothing$ do
      select and add the earliest emergency task
      $t_k \in Allow_{ji}^m$ to TO_j;
    else
      select and add a common task $t_k \in Allow_{ji}^m$
      to TO_j according to (7);
    end if
    $t_i \leftarrow t_k$;
    construct $Allow_{ji}^m$;
  end while
  $O \leftarrow O - o_j$;
end while

5.2 Repair strategy of emergency task requirement constraint

Time window constraints and sensor slewing switch time constraints are addressed in the pretreatment phase (Section 3) and problem modeling phase (Section 4). Uniqueness constraints, energy constraints and storage capacity constraints are tackled in the feasible solution construction stage (Section 5.1). To guarantee the satisfaction of the emergency task requirement constraint, a constraint repair strategy is designed, which is expounded as follows.

for $j = 1 \rightarrow |O|$ do
  for each $t_i \in TE_j$ do
    if $x_i = 0$ do
      insert t_i to TO_j in which t_i conflicts least with
      other tasks;
      delete the tasks conflicting with t_i from TO_j;
      set $x_i = 1$;
      end if
  end for
end for

The algorithm proposed above traverses every emergency task in TE_j and judges whether there are emergency tasks having not been scheduled yet. Each time when it finds out such an unscheduled emergency task t_i, it will insert t_i into the orbit in which t_i can be executed and conflicts least with other tasks. Then, the tasks conflicting with t_i will be deleted to eliminate conflicts caused by insertion of t_i.

5.3 Iteration local search

Iteration local search, whose behavior can be characterized as iteratively building a chain of solutions, is a conceptually simple meta-heuristic that nevertheless has led to state-of-the-art algorithms for many computationally hard problems [23]. ACO is an approximate algorithm based on constructive mechanism, whose neighborhood structure is different from local search mechanism used in iteration local search algorithms. Hence, we believe that ACO and iteration local search can supplement each other. That is to say, ACO can provide a better initial solution for iteration local search, and iteration local search, as a simple and well-performed meta-heuristic algorithm, can further optimize the solution obtained by ants efficiently.

Assume that s is the solution initially obtained by an ant, where $s = \bigcap_{j=1}^{|O|} TO_j$. The procedure of iteration local search optimizing s is described as follows.

$s^{\wedge} = \mathrm{BasicLocalSearch}\ (s)$;
$s_{\mathrm{best}} = s^{\wedge}$;
while s_{best} has improvement in every N iterations do

$s' = \text{Perturbation}\ (s^\wedge)$;
$s^* = \text{BasicLocalSearch}\ (s')$;
if $f(s^*) < f(s_{\text{best}})$ do
$s_{\text{best}} = s^*$;
end if
$s^\wedge = \text{AcceptanceRegular}\ (s^\wedge, s^*)$;
end while
return s_{best}

From the algorithm framework given above, $\text{BasicLocalSearch}(s)$ orderly selects a common task t_k from the unscheduled task set, whose elements are in descending order in terms of tasks' priorities. Then t_k is inserted into s without conflicts with any emergency task, and the common tasks conflicting with t_k from s are deleced. Thus, a new solution $s^\wedge$ is produced. Successively, $f(s)$ is compared with $f(s^\wedge)$. If $f(s) \leqslant f(s^\wedge)$, the current insertion is successful and the solution is updated. Otherwise, the insertion is failed and the update operation will be escaped. Then, the next task is selected to repeat the insertion operation until the unscheduled tasks set has been traversed completely.

$\text{Perturbation}\ (s^\wedge)$ generates perturbations in $s^\wedge$, which is an effective mechanism used to prevent the ACO-ILS from being convergent to local optimal solution too quickly. For each orbit o_j, it randomly selects an unscheduled common task t_k without conflicting with any emergency task to insert into TO_j, and deletes the common tasks conflicting with t_k from TO_j. Finally, it updates the solution no matter whether the solution has become worse or better.

$\text{AcceptanceRegular}\ (s^\wedge, s^*)$ decides to choose either $s^\wedge$ or s^* as the new initial solution for the next generation iteration. To strengthen the global exploitation ability of ACO-ILS, here we adopt a probability selection rule. Namely, select the better solution between $s^\wedge$ and s^* with the probability q $(0 < q < 1)$, and the worse with the probability $1 - q$.

Iteration local search conducts insertion or deletion operation just related to common tasks, therefore it produces no influence on emergency tasks and will not violate the emergency requirement constraint. The termination condition of iteration local search is that s_{best} has no improvement after N iterations.

5.4 Pheromone update rule

After every ant has found a feasible route, the pheromone on edges directed to common tasks should be updated according to the following formula:

$$\tau_{ik} = \begin{cases} \tau_{ik} + Q_L, & \text{rand}\ (0,1) < q_0 \\ \tau_{ik} + Q_G, & \text{otherwise} \end{cases} \tag{9}$$

where $Q_L = f(s_L)/p_{\text{all}}$, $Q_G = f(s_G)/p_{\text{all}}$, s_L is the current iteration optimization solution, $f(s_L)$ denotes the sum priorities of common tasks included in s_L, s_G is the current global optimization solution, $f(s_G)$ represents the sum priorities of common tasks included in s_G, and p_{all} denotes the sum priorities of all common tasks distributed to different orbits in the pretreatment phase.

Equation (9) implies that ACO-ILS will select the current iteration optimization solution to update pheromone with probability q_0, while select the current global optimization solution to update the pheromone with probability $1 - q_0$. This update strategy strives for a balance between exploitation and exploration ability of ACO-ILS.

After each iteration operation, pheromone evaporation operation should be carried out on related edges as well. The evaporation rule conforms to the following formula:

$$\tau_{ik} = (1 - \rho)\tau_{ik} \tag{10}$$

where ρ $(0 < \rho < 1)$ denotes the evaporation coefficient. To prevent the pheromone value from accumulating or evaporating too much, the pheromone should be adjusted according to the following strategy after each update operation:

$$\tau_{ik} = \begin{cases} \tau_{\max}, & \tau_{ik} > \tau_{\max} \\ \tau_{ik}, & \tau_{\min} \leqslant \tau_{ik} \leqslant \tau_{\max} \\ \tau_{\min}, & \tau_{ik} < \tau_{\min} \end{cases}. \tag{11}$$

5.5 Termination criterions of algorithm

Termination criterions are used to control the whole optimization process, which helps to get a balance between time consumption and solution quality. Two termination criterions are adopted here. The first one is that if the whole iteration time reaches Max_Iter, the algorithm terminates. The other one is that if the solution has no improvement after every Mid_Iter iteration operations, the algorithm terminates. The values of Max_Iter and Max_Iter should be provided at the beginning.

6. Experimental simulation and discussion

To analyze the performance of ACO-ILS, multiple cases have been designed for simulation. We first randomly generate four groups of point targets belonging to the area whose latitude is within the range of $-30°$ to $60°$ and longitude is within the range of 0 to $150°$. These groups include 100, 200, 300 and 400 targets, respectively. In each group, 95% of all targets are common tasks and the others are emergency tasks. Besides, six earth observation satellites are selected from STK database for simulation as well. Through the combination between these different satellites and groups of point targets, thirteen cases are prepared for the final simulation. The scheduling horizon is set as 24 h.

The priorities of common tasks is uniformly distributed among $[1, 10]$, and both energy consumption and time consumption are uniformly distributed among $[2, 6]$. Besides, the maximal storage capacity and energy capacity that could be used in each orbit are assumed to be 30 and 50, respectively. The average energy consumption for sensor slewing is $0.3°$.

Through the intensive experimental simulation, parameter values of the proposed algorithm are configured properly, which are listed in Table 1. The ACO-ILS is implemented by Matlab, and runs within a computer with AMD Turion 64×2, 1.61 GHz, and 512 M memory. Each case is computed for 20 times, and the average computational results are shown in Table 2. The time-window and slew angel for each task in each orbit are computed by STK tool before scheduling.

Table 1 Parameter values of ACO-ILS

Parameter	Value	Parameter	Value
AntSize	10	q	0.9
α	2.5	q_0	0.2
β	1.5	ρ	0.06
λ	0.5	$\tau_{\max}$	1.0
ω	0.3	$\tau_{\min}$	0.01
υ	0.2	Max_Iter	200
τ_0	0.2	Mid_Iter	10

Table 2 Computation results of cases simulation

N	N_p	N_s	TW	EN_p	$\overline{P}$	$\overline{t}$	$\overline{et}$	$\overline{T}(s)$
1	100	2	126	84	409	72	4	12.31
2		3	179	86	441	78	5	22.53
3		4	227	92	474	92	5	38.44
4	200	2	266	134	674	105	6	35.67
5		3	399	176	741	128	8	56.61
6		4	541	179	909	159	9	86.14
7	300	2	378	202	867	145	9	69.52
8		3	578	250	1 114	204	12	101.91
9		4	799	285	1 355	242	15	156.52
10		6	1 242	286	1 442	267	15	243.73
11	400	2	503	244	1 208	198	12	175.91
12		4	1 006	381	1 787	302	15	301.65
13		6	1 545	382	1 914	335	18	472.74

In Table 2, NC represents the index of each case, N_p is the number of targets, N_s denotes the number of satellites, TW is the number of available time-windows between all satellites and point targets, EN_p is the number of targets owing the available time-window, $\overline{P}$ is the average profit generated by finished common tasks, $\overline{t}$ is the average number of finished common tasks, $\overline{et}$ is the average number of finished emergency tasks, and $\overline{T}$ denotes the average time consumption.

From the analysis of Table 2, we discover that ACO-ILS could deal with the multi-satellite observation scheduling problem of each case within desired time, while the consumption time increases in an approximately linear trend. Especially, we observe that the ACO-ILS can resolve the problem about 400 tasks and 6 satellites within 500 s and get a feasible solution of high quality. Consequently, ACO-ILS could be considered as an efficient algorithm to resolve a multi-satellite observation scheduling problem including both emergency tasks and common tasks.

Wolfe and Stephen [24] compared the capability of dispatch algorithm, look-ahead algorithm and GA when attempting to solve multi-satellite observation scheduling problems. They drew conclusions that GA was most efficient in dealing with more complex problems. Similarly, Globus et al. [17] employed stochastic climbing hill, SA, GA, iterated sampling, and squeaky wheel optimization to resolve multi-satellite observation scheduling problems, respectively, and their experimental results demonstrated that SA outperformed the other algorithms entirely.

GA and SA have shown a good performance in resolving combination optimization problems including the satellite observation scheduling problem referred above. To compare the performance of ACO-ILS with GA and SA, we attempt to employ GA and SA to resolve multi-satellite observation scheduling problems as well. These three algorithms all adopt pretreatment operation and guarantee the execution of emergency tasks preferentially, so computation results about emergency tasks obtained by different algorithms are the same. Hence, it is only necessary to compare the computation results of common tasks. Similarly, each case is computed for 20 times by GA and SA.

The comparison of average computational results of these three algorithms is plotted in Fig. 4. From Fig. 4, we could find that ACO-ILS exhibits the best performance in each case, GA is worse than ACO-ILS, and SA performs worst. The reason is that, ACO-ILS is based on the acyclic directed graph model, which reflects the natural process of satellite observation activity. In addition, ACO really specializes in the complex problem that can be easily described by graph model, which has been demonstrated when solving the traveller sale problem and dynamic routing problem [22]. The satellite's observation process among different tasks is just similar to the process of ant's route searching. Furthermore, the introduction of iteration local search to ACO-ILS greatly quickens the solving process and improves solution quality.

To validate the advantage of the proposed integrated scheduling strategy compared with the traditional two-phased scheduling strategy, we conduct related comparison experiments. As referred in Section 1, the two-phased scheduling method first assigns satellite resources and time-windows to emergency tasks according to a greedy rule, then assigns the remain observation resources and time-windows to common tasks using ACO-ILS for the comparison fairness. Similar to the comparison among ACO-ILS, GA and SA, here it only needs to compare the computational results of common tasks as well.

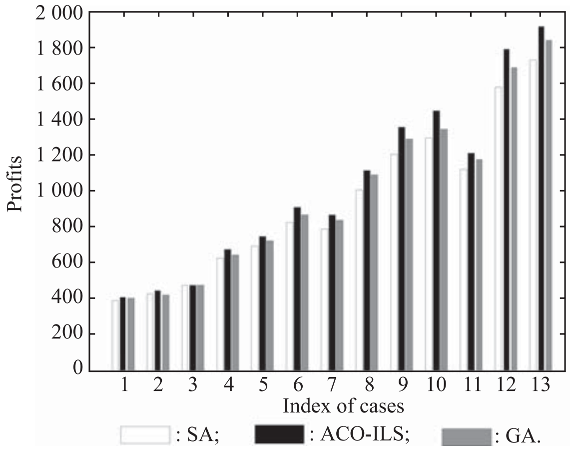


Fig. 4 Comparison among ACO-ILS, GA and SA

The corresponding comparison results are shown in Fig. 5, from which we can observe that the integrated scheduling strategy finds better scheduling solutions nearly under every case. When the scheduling problem tends to be more complex, such as more observation tasks, the superiority of the integrated scheduling strategy appears to be more obvious. That is because the integrated scheduling strategy attempts to synthetically arrange emergency tasks and common tasks to corresponding time-windows and orbits. It focuses on meeting emergency task requirements meanwhile maximizing the profit generated by common tasks. By contrast, the two-phase scheduling strategy assigns observation resources to emergency tasks firstly, which does not adequately consider the influence on latter

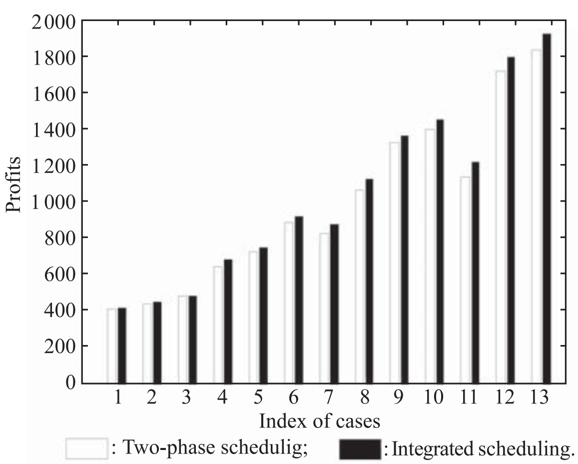


Fig. 5 Comparison between two-phase scheduling strategy and integrated scheduling strategy

execution of common tasks and may do harm to the profit of common tasks.

ACO is integrated with iteration local search when resolving the problem. To test its efficiency, ACO-ILS and the traditional ACO without iteration local search have been compared in this paper. Fig. 6 displays how the global optimization solution evolving with time when solving case 13 using ACO-ILS and ACO, respectively. From Fig. 6, we find that ACO-ILS has more powerful searching capability and can acquire solution with higher quality. On the contrary, ACO is prone to convergent to a local optimal solution more quickly. The reason is that, in ACO-ILS, iteration local search not only can speed up the solving process but also improve the solution quality rapidly.

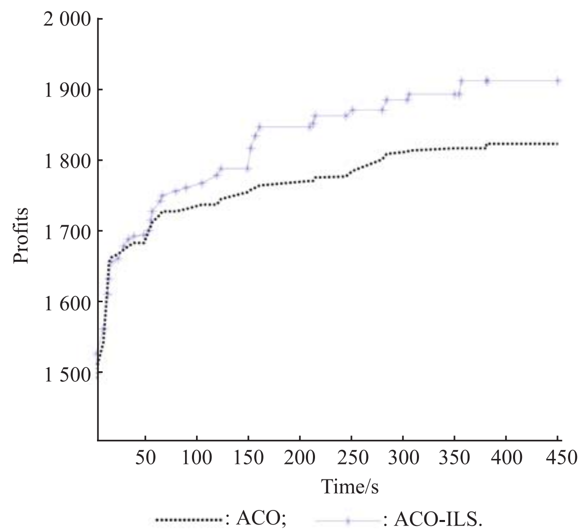


Fig. 6 Comparison between ACO-ILS and ACO

7. Conclusions and future work

Aiming at the real-world trace, we develop an integrated scheduling framework in order to solve the multi-satellite observation scheduling problem including both common tasks and emergency tasks simultaneously. In the framework, a hybrid ACO-ILS is proposed, which is based on an acyclic directed graph model and uses the iteration local search method to further improve the solutions initially obtained by ants. Extensive experimental simulation validates the satisfactory performance of ACO-ILS. Finally, through the comparisons and analysis of computational results, we find that the integrated scheduling strategy is more efficient than the two-phased scheduling strategy, and ACO-ILS outperforms both GA and SA.

The future work in our study includes three aspects:

(i) To make the schedule conform to users' demands more consistently, the fuzzy and uncertain requirements should be taken into account in the future satellite observation scheduling.

(ii) One satellite's observation belts may include multiple targets simultaneously, thus if multiple tasks satisfy certain slew angle constraints and time-window constraints, they can be combined into one task and accomplished via one observation activity of a satellite. Considering task clustering in satellite observation scheduling process can save resource and improve observation efficiency. Consequently, it is meaningful to take it into account when developing the satellite observation scheduling method.

(iii) Multiple kinds of satellite platforms with different observation mechanisms coexist in observation systems to accomplish various tasks together. It is crucial to make these observation platforms cooperate with effectively so as to enable a quick response to disturbances, uncertainties, and diversity of users' requirements. Consequently, an effective cooperative scheduling method for different kinds of observation resources can enhance the flexibility, robustness and adaptation of the whole observation system.

Acknowledgements

We thank Yinlin Li, Xiaomin Zhu, and Jianjiang Wang for their contributions to this study. We are also thankful to two anonymous referees.

References

[1] G. Nicholas, G. Hall, J. Michael. Maximizing the value of a space mission. *European Journal of Operational Research*, 1994, 78(2): 224–241.

[2] W. Potter, J. Gasch. A photo album of earth: scheduling Landsat 7 mission daily activities. *Proc. of the International Symposium on Space Mission Operations and Ground Data Systems*, 1998.

[3] E. Bensana, G. Verfaillie, J. C. Agnese, et al. Exact and inexact methods for the daily management of an earth observation satellite. *Proc. of the International Symposium on Space Mission Operations and Ground Data Systems*, 1996: 507–514.

[4] M. Vasquez, J. K. Hao. A "Logic-Constrained" knapsack formulation and a tabu algorithm for the daily photograph scheduling of an earth observation satellite. *Journal of Computational Optimization and Applications*, 2001, 20(7): 137–157.

[5] M. Vasquez, J. K. Hao. Upper bounds for the SPOT 5 daily photograph scheduling problem. *Journal of Combinatorial Optimization*, 2003, 7(1): 87–103.

[6] S. Harrison, M. Price, M. Philpott. Task scheduling for satellite based imagery. *Proc. of the 18th Workshop of the UK Planning and Scheduling Special Interest Group*, 1999: 64–78.

[7] W. Lin, D. Liao, C. Liu, et al. Daily imaging scheduling of an earth observation satellite. *IEEE Tran. on System, Man, and Cybernetics—Part A: Systems and Humans*, 2005, 35(2): 213–223.

[8] V. Gabrel, C. Murat. Mathematical programming for earth observation satellite mission planning. T. A. Ciriani, G. Fasano, S. Gliozzi, et al. *Operations Research in Space and Air*. Dordrecht: Kluwer Academic Publishers, 2003: 103–122.

[9] V. Gabrel, D. Vanderpooten. Enumeration and interactive selection of efficient paths in a multiple criteria graph for scheduling an earth observing satellite. *European Journal of Operational Research*, 2002, 139(3): 533–542.

[10] J. Wang, N. Jing, J. Li, et al. A multi-objective imaging scheduling approach for earth observing satellites. *Proc. of the 9th Annual Conference on Genetic and Evolutionary Computation*, 2007: 2211–2218.

[11] M. A. A. Mansour, M. M. Dessouky. A genetic algorithm approach for solving the daily photograph selection problem of the SPOT5 satellite. *Computers & Industrial Engineering*, 2010, 58(3): 509–520.

[12] N. Bianchessi, J. F. Cordeau, J. Desrosiers, et al. A heuristic for the multi-satellite, multi-orbit and multi-user management of earth observation satellites. *European Journal of Operational Research*, 2007, 177(2): 750–762.

[13] J. Frank, A. Jósson, R. Morris, et al. Planning and scheduling for fleets of earth observing satellites. *Proc. of the 6th International Symposium on Artificial Intelligence, Robotics, Automation and Space*, 2000.

[14] M. Lemaitre, G. Verfaillie, F. Jouhaud, et al. How to manage the new generation of agile earth observation satellites. *Proc. of the 6th International Conference on Space Operations*, 2000.

[15] D. S. Qiu, L. N. Zhang, J. H. Zhu, et al. FFFS-DTMB and ADTPC-DTMB algorithmin multi-satellites mission planning. *Acta Aeronautica et Astronautica Sinica*, 2009, 30(11): 2178–2184. (in Chinese)

[16] G. Verfaillie, M. Lemaitre. Tutorial on planning activities for earth watching and observation satellites and constellations: from off-line ground planning to on-line on-board planning. *Proc. of International Conference on Automated Planning and Scheduling*, 2006.

[17] A. Globus, J. Crawford, J. Lohn, et al. A comparison of techniques for scheduling fleets of earth observing satellites. *Proc. of the 16th Innovative Application of Artificial Intelligence Conference*, 2004.

[18] M. Dorigo. *Optimization, learning and natural algorithms*. Milano: Politecnico di Milano, 1992.

[19] T. Stützle, H. H. Hoos. Max–min ant system. *Future Generation Computer Systems*, 2000, 16(8): 889–914.

[20] G. Fuellerer, K. F. Doerner, R. F. Hartl, et al. Ant colony optimization for the two-dimensional loading vehicle routing problem. *Computers & Operations Research*, 2009, 36(3): 655–673.

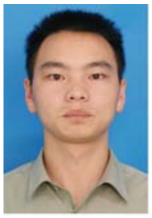
[21] C. Rajendran, H. Ziegler. Ant-colony algorithms for permutation flowshop scheduling to minimize makespan/total flowtime of jobs. *European Journal of Operational Research*, 2004, 155(2): 426–438.

[22] M. Dorigo, T. Stützle. *Ant colony optimization*. Cambridge: MIT Press, 2004.

[23] H. R. Lourenco, O. C. Martin, T. Stützle. Iterated local search: framework and applications. M. Gendreau, J. Potvin. *Handbook of Metaheuristics*. New York: Springer, 2010: 363–397.

[24] J. Wolfe, S. E. Stephen. Three scheduling algorithms applied to the earth observing systems domain. *Management Science*, 2000, 46(1): 148–168.

Biographies



Guohua Wu was born in 1986. He received his B.S. and M.S. degrees in operations research from National University of Defense Technology (NUDT), China, in 2008 and 2010, respectively, and currently he is a Ph.D. candidate in NUDT. His research interests are combat modeling and simulation, intelligent computing, combinatorial optimization, and space resource planning and scheduling technology.
E-mail: guohuawu@nudt.edu.cn

Manhao Ma was born in 1974. He received his M.S. and Ph.D. degrees in management science and technology from National University of Defense Technology (NUDT), China, in 2004 and 2008, respectively. He is currently an associate professor in School of Information Systems and Management at NUDT. His research interests include C4ISR systems, distributed simulation, combinatorial optimization, and space-based information systems.
E-mail: mhma@sina.com

Jianghan Zhu was born in 1972. He received his M.S. and Ph.D. degrees in management science and technology from National University of Defense Technology (NUDT), China, in 2000 and 2004, respectively. He is currently a professor and doctoral supervisor in School of Information Systems and Management in NUDT. His research interests include combinatorial optimization, space-based information systems, and satellite application information link.
E-mail: zhujianghan@nudt.edu.cn

Dishan Qiu was born in 1958. He is currently a professor and doctoral supervisor in School of Information Systems and Management in National University of Defense Technology. His research interests include combat modeling and simulation, combinatorial optimization, space-based information systems, and satellite application information link.
E-mail: ds_qiu@sina.com