***I.J. Image, Graphics and Signal Processing*, 2010, 1, 10-18**
Published Online November 2010 in MECS (http://www.mecs-press.org/)

Earth Observation Satellites Scheduling Based on Decomposition Optimization Algorithm

Feng Yao
College of Information Systems and Management, National University of Defense Technology, Changsha, China
Email:yaofeng@nudt.edu.cn

Jufang Li
College of Information Systems and Management, National University of Defense Technology, Changsha, China
Email:lijufang@nudt.edu.cn

Baocun Bai
College of Information Systems and Management, National University of Defense Technology, Changsha, China
Email:bcbai@nudt.edu.cn

Renjie He
College of Information Systems and Management, National University of Defense Technology, Changsha, China
Email:renjiehe@nudt.edu.cn

***Abstract*—A decomposition-based optimization algorithm was proposed for solving Earth Observation Satellites scheduling problem. The problem was decomposed into task assignment main problem and single satellite scheduling sub-problem. In task assignment phase, the tasks were allocated to the satellites, and each satellite would schedule the task respectively in single satellite scheduling phase. We adopted an adaptive ant colony optimization algorithm to search the optimal task assignment scheme. Adaptive parameter adjusting strategy and pheromone trail smoothing strategy were introduced to balance the exploration and the exploitation of search process. A heuristic algorithm and a very fast simulated annealing algorithm were proposed to solve the single satellite scheduling problem. The task assignment scheme was valued by integrating the observation scheduling result of multiple satellites. The result was responded to the ant colony optimization algorithm, which can guide the search process of ant colony optimization. Computation results showed that the approach was effective to the satellites observation scheduling problem.**

***Index Terms*—Earth Observation Satellites, decomposition, adaptive ant colony optimization, heuristic algorithm, very fast simulated annealing**

I. Introduction

Earth Observation Satellite is a kind of satellite which uses its optical or other remote sensing imaging equipment to observe ground targets and generates corresponding images. The satellite's routing job is controlled by uplink instructions which are transformed from schedules generated by the ground systems. Imaging satellites scheduling means to assign satellite resource and execution time to imaging tasks under the satellite operational constraints in order to maximize the number of tasks that could be completed [1].

The satellite imaging process can be shown as figure 1. Restricted by its orbit, a satellite will pass a ground target for limited times, and imaging operation must be executed within time windows that the satellite can catch sight of the target. For a target which is deviated from the ground trace of the satellite, the satellite or the sensor onboard must roll itself to aim at it. If the roll angles for different targets are not the same, the satellite must switch its attitude between different imaging activities, thus requiring enough transition time left. For example, there is not enough transition time between imaging of target B and target C, so only imaging target B after target A or imaging target C after target A is possible.

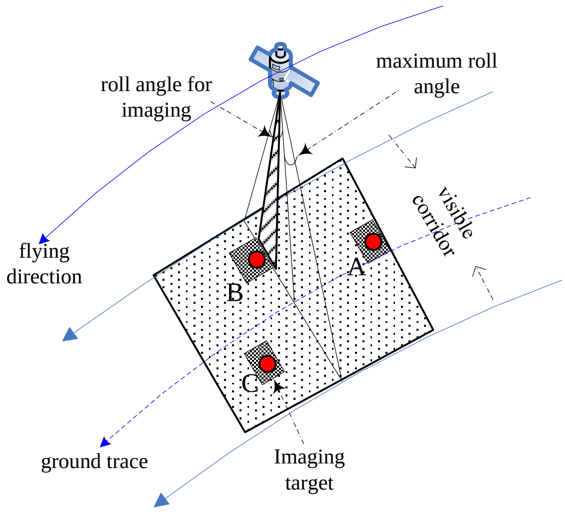

Figure 1. Satellite Imaging Process

Therefore, scheduling of a imaging satellite can be regarded as a single machine scheduling problem with time window constraint and setup time correlative with task order[2]. The problem was proved to be NP-hard[3],

Corresponding author: Feng Yao.

and was solved mainly by heuristic algorithm[4,5], neighborhood search algorithm[6-8], and meta-heuristics such as genetic algorithm[9].

As the number of imaging satellite continually increased, multi-satellites imaging scheduling became a research hotspot[1,5,7]. Multi-satellites imaging scheduling is more difficult than single satellite scheduling for the following reasons. First, the number of alternative satellites and alternative time windows for a imaging task is multiplied, making the combinational characteristic of the problem more prominent. Second, operational constraints of different satellites may be different, making uniform modeling of the problem difficult. To cope with these difficulties, most research made some simplifications to abstract those common main constraints and constructed mix integer programming model[5] or constraint satisfaction problem models[1,7]. Then approximate algorithms were used to solve the models. Such methods were effective in some cases, but when problem scale was large, computation time would be too long or computation result would be relatively poor.

In research on the problem of cooperative earth observing by multiple satellites, Morris[1] proposed an idea which first assigned observation tasks to different satellites, then solved single satellite scheduling problem for each satellite. The essence of his method was to transform multi-satellites scheduling problem into single satellite scheduling problem to decrease problem complexity and improve computational efficiency. Its task assignment process lied in a higher level and would greatly affect the scheduling effect, since task assignment process divided the tasks into several non-intersecting subsets and dissevered the relations among them. But how to find a good task assignment strategy to ensure the method's performance is itself a quite difficult problem. Morris didn't give a clear answer for it.

This paper used Morris' decomposition idea as a reference and decomposed the imaging satellites scheduling problem into a task assignment main problem and a single satellite scheduling sub-problem. The difference was that, Morris' method assigned tasks to satellites only once according to some strategy, while our method deemed task assignment as an optimization process also, and used an adaptive ant colony optimization algorithm (ACO) to generate different task assignment solutions. Based on these solutions and the following single satellite scheduling processes, we could get different complete schedules. Then assessment results of these schedules were fed back to ant colony optimization algorithm to guide it to find better task assignment solutions and complete schedules. Tests on different scale problems showed that our algorithm was effective.

II. Modeling of Imaging Satellites Scheduling Problem

Imaging satellites scheduling problem can be described as a quintuple $<E,S,T,C,F>$. E is scheduling horizon. $S=\left\{S_1,\cdots,S_{N_S}\right\}$ is a set of satellites. $T=\left\{T_1,\cdots,T_{N_T}\right\}$ is a set of tasks. C is a set of constraints. F is objective function. Imaging satellite scheduling can be classified into long-term scheduling and short-term scheduling. Short-term scheduling as this paper considered requires precise orbit prediction data, so its horizon is usually shorter than 24 hours.

Due to a imaging satellite's field of view, it can see only a strip on the earth at one pass, so big area must be segmented to narrow strips along with the satellite's ground trace. For multi-satellites case, different satellite's ground traces may be not parallel, so segmenting target areas into small spots that can be observed completely by either satellite in one pass is a feasible way. For simplification, we assumed that all the observing targets were spot targets in this paper.

Each observing task must satisfy time interval, sensor type, ground sample distance or other requirements. For a given task, there may be more than one satellite that can fulfill it. Each candidate satellite may have several time windows with the task in a given time period. Each time window means a specific roll angle for the corresponding satellite. These time windows and roll angles can be obtained ahead of schedule through orbit calculation and prediction software.

For a task T_i , we define all of its time windows as a set

$$O_i=\bigcup_{j=1}^{N_S}\bigcup_{k=1}^{N_{ij}}O_{ijk},i\in[1,\cdots,N_T],$$

where N_T is the number of tasks, N_S is the number of satellites, N_{ij} is the number of time windows between satellite S_j and task T_i , O_{ijk} is the k*th* time window. g_{ijk} is the roll angle of S_j observing T_i in O_{ijk} . $[ws_{ijk},we_{ijk}]$ is the corresponding time interval of O_{ijk} . p_i is the priority of T_i . d_i is the imaging duration for T_i .

For a satellite S_j , its rolling velocity is r_j , the stabilizing time needed after rolling is h_j , the memory needed for unit duration time of imaging is α_j , the maximum storage capacity is M_j , the maximum times of imaging operations in the given scheduling horizon is R_j . x_{ijk} is the decision variable which is defined as

$$x_{ijk}=\begin{cases}1, & \text{task } T_i \text{ is scheduled to satellite } S_j \text{in time window} O_{ijk}\\ 0, & \text{else}\end{cases}$$

.

For imaging satellites scheduling problem, the following constraints must be satisfied.

(1) Exclusivity, namely each task should be scheduled only once and without interrupt, i.e.

$$\sum_{j=1}^{N_S}\sum_{k=1}^{N_{ij}}x_{ijk}\le 1,\forall i\in[1,\cdots,N_T] \tag{1}$$

(2) There must be enough time left for a satellite to switch its roll angle between two successive imaging operations. This time includes transition time and the stabilizing time after a switch, i.e.

$$\begin{aligned} & we_{ijk} + | g_{ijk} - g_{i'jk'} | / r_j + h_j \le ws_{i'jk'} \\ & \forall j \in [1,\cdots,N_S], x_{ijk} x_{i'jk'} = 1 \text{且} we_{ijk} \le ws_{i'jk'} \end{aligned} \tag{2}$$

(3) The amount of image data stored onboard should not overstep the capacity of memory instrument, i.e.

$$\sum_{i=1}^{N_T} \sum_{k=1}^{N_{ij}} x_{ijk} d_i \alpha_j \le M_j, \forall j \in [1,\cdots,N_S] \tag{3}$$

(4) The times of imaging operations are limited due to the satellite's power and maneuverability constraints, i.e.

$$\sum_{i=1}^{N_T} \sum_{k=1}^{N_{ij}} x_{ijk} \le R_j, \forall j \in [1,\cdots,N_S] \tag{4}$$

To sum up, the imaging satellites scheduling problem discussed in this paper is to assign satellite resources and imaging time to tasks subjected to the above constraints, with an optimization objective of maximizing the sum of all the scheduled tasks' priorities. Thereby, we establish a 0-1 integer programming model of the problem as follows.

$$\begin{aligned} & \max \sum_{i=1}^{N_T} \sum_{j=1}^{N_S} \sum_{k=1}^{N_{ij}} x_{ijk} p_i \\ & s.t. \begin{cases} \sum_{j=1}^{N_S} \sum_{k=1}^{N_{ij}} x_{ijk} \le 1 \\ we_{ijk} + d_i + | g_{ijk} - g_{i'jk'} | / r_j + h_j \le ws_{i'jk'}, \\ \quad \forall x_{ijk} x_{i'jk'} = 1 \text{且} we_{ijk} \le ws_{i'jk'} \\ \sum_{i=1}^{N_T} \sum_{k=1}^{N_{ij}} x_{ijk} d_i \alpha_j \le M_j \\ \sum_{i=1}^{N_T} \sum_{k=1}^{N_{ij}} x_{ijk} \le R_j \\ i, i' \in [1,\cdots,N_T] \\ j \in [1,\cdots,N_S] \\ k, k' \in [1,\cdots,N_{ij}] \end{cases} \end{aligned} \tag{5}$$

III. Framework of Solving Method

Considering the computational complexity of the above model, we propose a solving method based on problem decomposition. The original problem is decomposed into a task assignment main problem and a single satellite scheduling sub-problem. Its framework can be shown as figure 2.

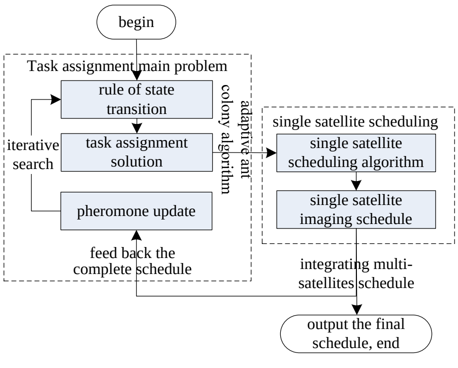
begin
Task assignment main problem
rule of state transition
task assignment solution
pheromone update
iterative search
adaptive ant colony algorithm
single satellite scheduling
single satellite scheduling algorithm
single satellite imaging schedule
feed back the complete schedule
integrating multi-satellites schedule
output the final schedule, end

Figure 2. Framework of Decomposition-Based Solving Method

In figure 2, the task assignment main problem considers a set of tasks $T = \{T_1,\cdots,T_{N_T}\}$ and a set of satellites $S = \{S_1,\cdots,S_{N_S}\}$. Task T_i has a set of alternative satellites $S(i)(S(i) \subseteq S)$ which not only can satisfy its imaging requirements such as ground sample distance, sensor type, etc., but also have time windows with it. The problem requires assigning a satellite to each task, and each task should be assigned to not more than one satellite.

Given a set of tasks $T^{'}$ which are assigned to satellite S_j, the single satellite scheduling sub-problem requires optimally selecting and scheduling tasks under the satellite operational constraints in order to maximize the number of tasks that could be completed.

The framework adopts adaptive ant colony optimization algorithm to solve task assignment main problem. In the iterative searching process, as soon as a task assignment solution is generated by the ants, it is sent to each corresponding satellite to carry on single satellite scheduling. The schedules from each satellite are integrated to obtain a complete schedule and assess performance of the task assignment solution. These assessment results are then used to guide ant colony optimization algorithm to find better task assignment solutions. After a number of iterations, a final optimized task assignment solution and corresponding multi-satellites imaging schedule is acquired. In such a framework, the single satellite scheduling module can be regarded as the attractiveness calculation function in ant colony optimization algorithm.

IV. Adaptive ACO Algorithm for Task Assignment Main Problem

Ant colony optimization algorithm is originated from the swarm behavior of ants searching for food [10]. It has already been successfully applied in a serial of difficult combinatorial optimization problems. Since its positive feedback characteristic and random searching ability are outstanding, we construct an adaptive ant colony optimization algorithm for the task assignment main problem.

A. *Definition of Pheromone and Its Initialization*

In TSP applications, pheromone often denotes the probability of getting better route if node j is arranged right after node i. Similarly in our algorithm, we define pheromone as the probability of assigning a certain task to a certain satellite, which is in accordance with the information that assigning that task to that satellite tends to get a better schedule. Let $|S(i)|$ be the number of alternative satellites for task T_i. The probability of assigning T_i to satellite S_j is defined as τ_{ij}. As initialization, the probability of assigning a task to each candidate satellite is set to be identical, i.e.

$$\tau_{ij}(0) = \frac{1}{|S(i)|}, \; i=1,\cdots,N_T, j=1,\cdots,|S(i)| \tag{6}$$

B. Construction of Solutions

When ants construct solutions, an ideal manner is to get feasible solutions at any time, meaning that tasks are assigned under all the constraints. But computation time needed in this manner is too long since imaging satellite's operational constraints are very complex. Besides, searching randomicity of ant colony optimization algorithm will be affected if feasible solution is required each time, thus the algorithm's performance will be degenerated[11]. In order to avoid this defect, we assume to consider only the constraint (1) as shown in section II when ants constructing solutions, other constraints are left for single satellite scheduling module to cope with. So in solution construction phase, we only need to select a certain satellite for a given task from its alternative satellites according to stated rule of state transition.

We adopt a pseudo-random proportion rule[10] to select satellite for a task. Let $\sigma_i = \sigma_i^{j_0}$ denote that task T_i choose satellite j_0 to fulfill it. The rule for making a choice is as follows:

$$\sigma_i = \begin{cases} \max\limits_{j\in[1,|S(i)|]}\{\tau_{ij}(t)\}, \text{if} \quad q \le q_0 \\ \phi, \qquad\qquad \text{otherwise} \end{cases} \tag{7}$$

q is a random number distributed uniformly in $[0,1]$, q_0 is a control parameter for random selection, $0 \le q_0 \le 1$. When $q \le q_0$, satellite is selected according to amount of pheromone. When $q > q_0$, ϕ means to select satellite according to the following probability distribution:

$$\Pr(i, j_0, t) = \frac{\tau_{ij_0}(t)}{\sum\limits_{j\in[1,|S(i)|]} \tau_{ij}(t)}, i \in [1, N_T], j_0 \in [1, |S(i)|] \tag{8}$$

$\Pr(i, j_0, t)$ denotes the probability of select satellite j_0 for T_i at time t.

C. Pheromone Update

1) Local Update Rule

Every time a solution Sol is constructed by the ants, it will be sent to single satellite scheduling module, then be further repaired (will be explained in section E) to get a new solution Sol'. If Sol' is better than Sol, pheromone would be accordingly updated locally. The update manner is as follows:

$$\tau_{ij}(t+1) = (1-\rho_{local})\tau_{ij}(t) + \rho_{local}\tau_{ij}(0), \forall \sigma_i^j \in Sol' \tag{9}$$

$\rho_{local} \in (0,1)$ is a local pheromone evaporation parameter. The local pheromone update rule can be seen as a diversification mechanism which can avoid getting into local optimum too early.

2) Global Update Rule

The purpose of global pheromone update is to reinforce pheromone of better solutions in order to guide the search more intelligent. Elitist strategy uses the best solution of each iteration to update pheromone, which tends to lead to early convergence[10]. Rank-based strategy ranks solutions of each iteration first, then updates pheromone according to weighted order information[12]. We adopt a best solution queue (BSQ) strategy to make global pheromone update. In BSQ, several current best solutions are always kept. If better solutions are got in iterating process, they will be replaced into BSQ. Each time an iteration is completed, BSQ will be updated, and pheromone will also be updated in term of the following mode:

$$\tau_{ij}(t+1) = (1-\rho_{global})\tau_{ij}(t) + \rho_{global}\Delta\tau_{ij} \tag{10}$$

$\rho_{global} \in (0,1)$ is a global pheromone evaporation parameter, while $\Delta\tau_{ik}$ is the increase amount of pheromone calculated as follows:

$$\Delta\tau_{ij} = \begin{cases} \dfrac{\lambda}{|S(i)|} \sum\limits_{r\in[1,|S(i)|]} \tau_{ir}(t), \text{ if } \sigma_i^j \in \text{BSQ} \\ 0, \qquad\qquad \text{otherwise} \end{cases} \tag{11}$$

λ is a pheromone increment factor.

D. Adaptive Strategy

Ant colony optimization algorithm bears a contradiction of quickly convergence and stagnant precocity. The strategies used to improve global search ability and avoid early getting in local optimum include, combining determinate selection with random selection, or change some parameters to dynamically adjust pheromone [13,14]. We adopt an adaptive parameter adjustment strategy to dynamically suite searching process.

Let parameter q_0 has a domain of $\{q_0^a, q_0^b\}$, $q_0^a < q_0^b$, parameter ρ_{global} has a domain of $\{\rho_{global}^a, \rho_{global}^b\}$, $\rho_{global}^a < \rho_{global}^b$. If the current best solution hasn't been improved after δ_1 times of successive iteration, we may suspect that the algorithm got in a local optimum, following search may not jump out or jump out too slowly. At this time, q_0 may be assigned a small value q_0^a, meaning high randomicity of satellite selection, in order to improve the possibility of jumping out local optimum and realizing a global search. At the same time, decreasing ρ_{global} may reduce corresponding pheromone of the current best solution, which will also help to get away from local optimum. After a local optimum is jumped out, the two parameters will be assigned big values q_0^b and ρ_{global}^b to improve converging speed.

Besides, since pheromone update is continuously, differences between pheromone may be remarkable, causing the search process easily got into stagnation. Only increasing the probability of random selection or change the rate of pheromone evaporation would not lead the search process out of local optimum. So, we adopt the idea of pheromone smoothing. When the current best solution was not improved for successive δ_2 times, each satellite's pheromone would be weighted averaged with the highest value among them. Such a method will minish the differences between pheromone when the algorithm get into stagnation, thus be propitious to lead it getting out of local optimum.

E. Solution Repair

The above algorithm presents a task assignment result each time it generates a solution. After the single satellite scheduling is finished, some tasks may not be arranged due to satellite capability limitation. We may try to adjust them to other satellites, meaning try to insert them into another satellite's schedule list. If an insertion is feasible, then the task assignment scheme is accordingly modified. This solution repair process can realize fine-tuning for the task assignment result and improve the solution quality. The detailed process is not complex and is omitted here.

V. THE SINGLE SATELLITE SCHEDULING SUB-PROBLEM

Single satellite scheduling can be looked upon as a single machine scheduling problem with time-window constraint and task order dependent set up time. Restricted by its orbit, a satellite has only limited imaging chances or visible time windows with a given target. Thus, single satellite scheduling with short horizon can be simplified to a task selection problem[8]. For small-scale single satellite scheduling problem, even complete searching algorithm can be used to get an optimal solution.

By our method, each time the ACO algorithm adopted here generates a solution, single satellite scheduling will be performed, so single satellite scheduling algorithm must be quick enough. We adopted a Heuristic Algorithm (HA) and a Very Fast Simulated Annealing algorithm (VFSA). Accordingly, we call our decomposition optimization method using HA algorithm as ACO-HA, while the method using VFSA as ACO-VFSA.

A. Heuristic Algorithm

HA selects optimal choices in turn according to some heuristic rules in its searching process, so it has very fast computation speed and the result is easy to explain. HA is an approximate algorithm in common use in satellite scheduling. Scheduling of ASTER, SPOT-5 and Landsat 7 satellite all use this kind of algorithm. Literature [4] provided some rules for task selection by synthetically considering task priority, sensor parameters and imaging quality. We consider conflicts between tasks except for task priorities, and define the concept of conflict degree for a task. If one task was scheduled, then conflict degree of that task would be the sum of the priorities of those not scheduled tasks impacted by the scheduled task. Let conflict degree of task T_i be $Conflict_i$, its calculation formula is:

$$Conflict_i = \sum_{T_j \in ConflictSet_i} p_j \qquad (12)$$

$ConflictSet_i$ is the set of tasks that could not be scheduled because of T_i's impact. Conflict degree reflects the degree of confliction between one task and the others. Based on this concept, our algorithm follows such a simple idea: schedule those tasks of small conflict with others firstly, with the expectation that more tasks will be scheduled. As figure 3 shows, there are conflicts between Task2 and Task1 or Task3, if we first schedule Task2, then Task1 and Task3 could not be scheduled both, so we should schedule Task1 or Task3 firstly.

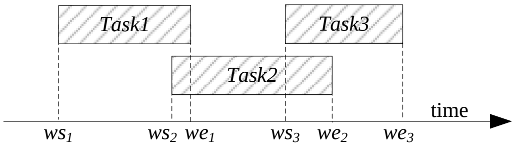

Figure 3. conflicts between tasks

Here, conflict between tasks mainly indicates time conflict, meaning that if there is not enough transition time between two tasks, then we take it for a conflict. The process of our HA algorithm is described as follows:

Step1: order the tasks according to task priorities, schedule the task with highest priority first.

Step2: for tasks with same priorities, calculate their conflict degrees, schedule the task with small conflict degree first.

Step3: for tasks with same priority and same conflict degree, schedule the task with small roll angle or early start time first.

B. Very Fast Simulated Annealing Algorithm

Heuristic algorithm requires in-depth comprehension of the problem itself. Since the algorithm selects task according to given rules and doesn't allow backtrack, its performance relies heavily on good heuristic rules and has no guarantee. To get better single satellite imaging schedule, we also try to use a simulated annealing algorithm. General simulated annealing algorithm has better global search ability, but need adequate perturbation and iteration numbers, with deliberate annealing plan, therefore its efficiency is still under improvement. As said before, in each iteration of our ACO algorithm, single satellite scheduling must be carried through for each satellite according to task assignment result. To avoid unacceptable time cost, single satellite scheduling must be quick enough. Ingber and some others improved perturbation and annealing manner for general SA and proposed a Very Fast Simulated Annealing algorithm [16, 17], which greatly increased the computation speed. We adopt the VFSA algorithm to solve our single satellite scheduling problem.

1) Solution encoding

Considering the neighborhood design, we arrange the tasks on each satellite according to the start time of their time windows, thus an activity list with time precedence is constructed. This method is convenient for setup time calculation and other constraint check to ensure new solution feasibility.

Besides, since satellite scheduling is usually an over-subscribed problem that only part tasks could be scheduled, we design a virtual satellite resource, which is used to virtually undertake all the tasks that could not be assigned to real satellites. The difference between virtual and real satellites is that all tasks assigned to virtual satellite are not restricted by time window or other constraints and their values are zero. Figure 4 is a denotation of solution encoding for a problem including 2 satellites and 10 tasks.

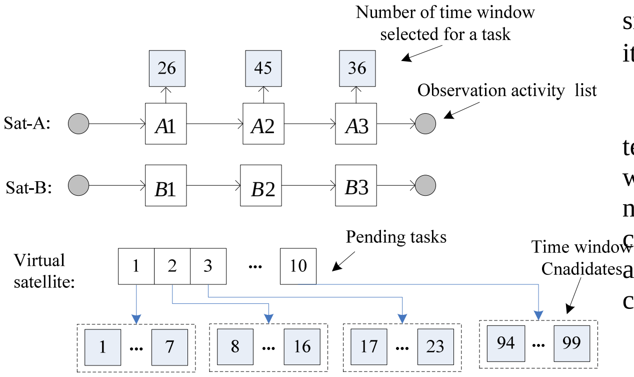

Figure 4. denotation of solution encoding

2) Model perturbation

VFSA adopts a pseudo-Cauchy distribution depending on temperature to perturb the current model and generate new solutions. We construct some neighborhood structures according to the problem feature. Moving between neighbors will perturb the original model and generate new solutions. Perturbations to a solution include task changes and task time changes.

For task change, we adopt the $\lambda - interchange$ method or named *Swap/shift move* method. In detail, we use the 1-0 node interchange and 1-1 node interchange method. 1-0 node interchange will insert a task assigned to one certain satellite to another satellite when feasible. 1-1 node interchange will swap the places of two tasks originally assigned to two different satellites when feasible. For single satellite scheduling, since there is only one real satellite and one virtual satellite, this method essentially means to swap tasks between real satellite and virtual satellite.

For task time change, we adopt an exchange neighborhood, namely randomly switch a task from one of its time window to another.

3) Acceptance probability

VFSA adopts such an acceptance probability formula according to generalized Boltzmann−Gibbs distribution:

$$P = \left[1-(1-q)\Delta E/T\right]^{1/(1-q)} \qquad (13)$$

Here, ΔE is energy difference, namely the difference between new solution and original one. q is a real number, different value of q means different preference of adventuring. Low value of q means that the acceptance probability will take a risk to make some bad solutions accepted more likely, while the computation speed will be improved, thus the value of q must be properly selected. When $q \to 1$, the acceptance probability will be just the same as it of general SA:

$$P = \exp(-\Delta E / T) \qquad (14)$$

4) Annealing plan

Since annealing speed is an important factor to decide efficiency of the algorithm, we adopt a fast annealing strategy:

$$T_k = T_0 \alpha^{k^{1/N}} \qquad (15)$$

Here, k is the number of iteration, α is the rate of temperature attenuation, N is a given parameter. The smaller N is, the faster temperature descends. Usually, it can be assigned a value of 1 or 2.

5) Terminate rules

To cut computation time, we adopt a dual-level terminate rule. The outer level is based on temperature, while the inner level is based on non-update of solution, namely when the current best solution is not improved for certain iterations, the search process will terminate. Our algorithm will terminate as long as either one of the conditions be satisfied.

VI. Experiments

Since there are still not acknowledged benchmark test problems in satellite scheduling research area, we use a random generation method to test our algorithm. The test problems are generated according to the following rules:

(1) Generate targets within an area between north latitude 20°~50° and east longitude 70°~130° according to a uniform distribution. Requirements of sensor type and ground sample distance are ignored. The number of targets M is respectively 100, 200, 300, 400 and 500.

(2) The number of satellites N is respectively 3, 4 and 5.

(3) The priorities of tasks are randomly selected among [1,10].

(4) The time windows between a task and a satellite are acquired by the Satellite Tool Kit software [18].

(5) Duration of a task imaging activity is 5~9 seconds.

(6) Different scale problem instances are generated through different combination of M and N.

Our algorithm was realized using C# and VS.net 2003 and was run on a workstation with configuration of Xeon(R) 5160 3.0 Hz CPU and 2G RAM.

Parameters of VFSA was set to q = -1， α =0.95， N =1. Parameters of ACO were set as table I shows.

TABLE I. PARAMETERS SET FOR ACO

parameters	meaning
$AntSize = 8$	number of ants
$MaxIter = 300$	max iteration number
$\rho_{local} = 0.1$	local pheromone evaporation coefficient
$\rho_{global}^{a} = 0.6, \rho_{global}^{b} = 0.8$	min and max value of global pheromone evaporation coefficient
$q_0^a = 0.55, q_0^b = 0.85$	min and max value of parameter denoting randomicity of satellite selection
$\delta_1 = 50$	the iteration number threshold for calling adaptive parameter adjustment strategy
$\delta_2 = 90$	the iteration number threshold for calling pheromone smooth strategy

Literature [7] constructed a constraint satisfaction problem model for multi-satellites scheduling problem, and solved the model using a tabu search algorithm. To test the performance of our algorithm, we compared our algorithms with it. The three algorithms were run 10 times for each problem instance, their avenue results which express the total revenue of the observed tasks are showed as table II.

From the results we can find that, in most cases, ACO-HA and ACO-VFSA are better than TS algorithm. Only for three instances 3, 5 and 6, improvements are not marked. By analyzing these problems and their solutions, we found that satellite resources in these problems are relatively sufficient, and the fulfillment rates of tasks are high. For example, in problem instance 3, when 100 tasks were to be scheduled on 5 satellites, all the tasks could be scheduled even using TS algorithm, so advantages of ACO-HA and ACO-VFSA would not be visualized.

Satellites scheduling problems are always over-subscribed problems, task requirements generally go beyond the satellites' ability, so our algorithm has more advantages. Besides, from table 2 we can also find that, the more complex our problem is, the better ACO-HA and ACO-VFSA are.

For ACO-HA and ACO-VFSA, we find that ACO-VFSA is better than ACO-HA. In each iteration, ACO-HA uses some heuristic rules to schedule tasks on single satellite, while ACO-VFSA uses very fast simulated annealing to schedule tasks on single satellite, which is able to get better single satellite schedule and correspondingly get better complete multi-satellites schedule.

As to computation time, we find that TS is faster for small scale problems, while with the increase of tasks and satellites, computation time required by TS goes up prominently. Since ACO−HA uses heuristic rules for single satellite scheduling, its performance is not evidently affected by the number of tasks and its computation time increases almost linearly. Therefore, ACO−HA is fitter for big scale problems with rigid time requirement. ACO−VFSA uses VFSA for single satellite scheduling in each iteration of ACO, so its time performance is affected by number of tasks more evidently. But even for the most complex test instance, it can return the result within 600s, which could be accepted in real applications.

TABLE II. COMPUTATION RESULTS OF EACH PROBLEM INSTANCE BY DIFFERENT ALGORITHMS

NO.	M	N	OBS	TS		ACO-HA			ACO-VFSA		
				RESULT	CPU(s)	RESULT	CPU (s)	GAP(%)	RESULT	CPU (s)	GAP (%)
1	100	3	289	475.3	4.21	486.6	25.06	2.3774	502.8	45.2	5.7858
2		4	384	514.2	5.64	518.8	35.77	0.8946	524.9	72.36	2.0809
3		**5**	**452**	**546.4**	**5.78**	**546.4**	**42.06**	**0**	**546.4**	**87.04**	**0**
4	200	3	521	715.7	25.65	735	48.4	2.6967	752.3	88.23	5.1139
5		**4**	**699**	**844.0**	**26.5**	**848.8**	**63.03**	**0.5687**	**855.7**	**106.98**	**1.3863**
6		**5**	**839**	**976.1**	**29.93**	**980.6**	**82.70**	**0.4610**	**985.2**	**153.26**	**0.9323**
7	300	3	802	841.9	43.5	856.2	70.44	1.6985	876.1	116.7	4.0622
8		4	1063	978.3	73.07	1009.4	92.53	3.1790	1019.8	127.5	4.2421
9		5	1296	1164.7	85.10	1204	123.65	3.3743	1221.4	188.46	4.8682
10	400	3	1074	862.0	120.70	903.6	84.32	4.8260	916.9	225.21	6.3689
11		4	1415	1022.1	155.75	1080.8	112.44	5.7431	1124.0	262.01	9.9697
12		5	1737	1236.4	173.43	1294.2	137.92	4.6749	1310.1	346.03	5.9609
13	500	3	1300	894.4	226.78	934.6	97.40	4.4946	952.6	477.89	6.5072
14		4	1716	1077.8	268.84	1119	145.2	3.8226	1144.2	502.81	6.1607
15		5	2123	1291.9	327.96	1395.8	183.92	8.0424	1452.6	581.53	12.4390

Note: NO. is the number of problem instances; OBS is the sum of all tasks' time windows predicted by STK in one instance; RESULT is the computation result; CPU is the computation time which is measured by second; GAP is the percentage of improvement when computation results are compared with TS algorithm.

Besides, in order to test the effect of adaptive strategy used in ACO, we solve the instance 15 using ACO−VFSA with and without adaptive strategy respectively, the evolvement of the best solution along with iteration number is showed as figure 5. It can be found that ACO−VFSA with adaptive strategy has stronger search ability and can approach best solution quicker. The algorithm with adaptive strategy got to optimum at the 77th iteration, while the algorithm without adaptive strategy got to a local optimum at the 121st iteration and couldn't improve the solution anymore.

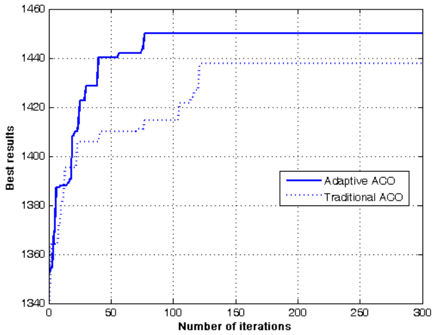

Figure 5. evolvement of the best solution for instance 15

VII. Conclusions

This paper proposed an optimization algorithm based on decomposition for multi-satellites observation scheduling problem. The problem was decomposed into task assignment main problem and single satellite scheduling sub-problem. We adopted an adaptive ACO algorithm to search optimized task assignment scheme while a heuristic algorithm and a VFSA algorithm was respectively used to solve the single satellite scheduling sub-problem. Results of single satellite scheduling were used to evaluate and guide the task assignment process. In task assignment procedure, only task exclusivity constraint was considered, in order to quickly construct a task assignment scheme. Other constraints were considered in single satellite scheduling procedure, in order to ensure feasibility of the solution. Finally, our algorithm was proved to have better efficiency through some random tests. ACO−VFSA could get better schedule while ACO−HA could return the final solution more quickly.

Besides, we found that our ACO−VFSA is notably affected by the problem scale. Some parameters also have direct impact on the computation result and efficiency. So how to select the parameters will be an important factor for improving our ACO−VFSA.

Acknowledgment

We thank anonymous referees for their constructive comments. This research is supported by National Natural Science Foundation of China under Grant 70801062.

References

[1] Robert A Morris, Jennifer L Dungan, John L. Bresina. An Information Infrastructure for Coordinating Earth Science Observations. In Proc. 2nd IEEE International Conference on Space Mission Challenges for Information Technology. 2006

[2] Wei-Cheng Lin, Da-Yin Liao, Chung-Yang Liu, Yong-Yao Lee. Daily Imaging Scheduling of an Earth Observation Satellite. IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART A: SYSTEMS AND HUMANS, vol. 35, pp.:213-223, 2005

[3] Bensana E, Verfaillie G, Bataillie N, Bluestein D. Exact and Approximate Methods for the Daily Management of an Earth Observing Satellite. Proceedings of SpaceOPS, Germany: Munich, 1996.

[4] XU Xue-ren, GONG Peng, HUANG Xue-zhi, Jin Yong. Study on Optimization Algorithms for Remote Sensing Date Collection Planning of Satellite. Journal of Remoter Sensing, vol. 56, pp. :962-968, 2007(In Chinese)

[5] Nicola Bianchessi, Jean-Francois Cordeau, Jacques Desrosiers, Gilbert Laporte, Vincent Raymond. A Heuristic for the Multi-Satellite, Multi-Orbit and Multi-User Management of Earth Observation Satellites. European Journal of Operational Research. vol. 177, pp. 750-762, 2005

[6] Cordeau J-F, Laporte G. Maximizing the Value of an Earth Observation Satellite Orbit. Journal of the Operational Research Society, vol. 56, pp. :962-968, 2005

[7] HE Ren-jie. Research on Imaging Reconnaissance Satellite Scheduling Problem. PhD thesis, Nation University of Defense Technology, 2004

[8] M. Lemaître, G. Verfaillie. Selecting and scheduling observations of agile satellites. Aerospace Science and Technology. 2002,(6) :367-381

[9] Wolfe W.J, Sorensen S.E. Three Scheduling Algorithms Applied to the Earth Observing Systems Domain. Management Science, vol. 46, pp. 148~168, 2000

[10] Dorigo M, Stutzle T. Ant Colony Optimization. Cambridge, MA: MIT Press, 2004

[11] Christian Blum, Ant colony optimization: Introduction and recent trends. Physics of Life Reviews, vol. 2, pp. :353–373, 2005

[12] B. Bullnheimer, R. R Hard, C. Strauss. A new rank-based version of the Ant System: A computational study. Central European Journal for Operations Research and Economics, vol. 7, pp. :25-38, 1999

[13] ZHU Qing-Bao, YANG Zhi-Jun. An Ant Colony Optimization Algorithm Based on Mutation and Dynamic Pheromone Updating. Journal of Software, vol. 12, pp. 185-192, 2004

[14] WANG Ying, XIE Jian–ying. An Adaptive Ant Colony Optimization Algorithm and Simulation. JOURNAL OF SYSTEM SIMULATION, vol. 14, pp. 31-33, 2002

[15] Stutzle T, Hoos H H. MAX-MIN ant system. Future Generation Computer Systems, vol. 16, pp. 889-914, 2000

[16] Ingber L. Very Fast Simulated Re-annealing. Math Compute Modelling, vol. 12, pp. :967-973, 1989

[17] CHEN Huagen, LI Lihua, XU Huiping, CHEN Bing. Modified Very Fast Simulated Annealing Algorithm. JOURNAL OF TONGJI UNIVERSITY(NATURAL SICENCI), vol. 34, pp. 1121-1125, 2006

[18] Analytical Graphics Inc. Satellite Tool Kit 6.0. http://www.agi.com

Feng Yao was born in Xianyang, Shanxi Province, China in 1978. Yao received the B.S. degree from National University of Defense Technology in 2003. He is now a lecturer at National University of Defense Technology. His research interest covers mission planning, scheduling and system optimization.

Jufang Li was born in Linzhou, Henan Province, China in 1978. Li received the Ph.D. degree from National University of Defense Technology in 2004. She is now a associate professor at National University of Defense Technology. Her research interest covers mission planning, scheduling and intelligent optimization algorithms.

Baocun Bai was born in Cangzhou, Hebei Province, China in 1981. Bai received the Ph.D. degree from National University of Defense Technology in 2009. His research interest covers spacecraft mission planning and combinational optimization.

Renjie He was born in Yuncheng, Shanxi Province, China on September 1976. He received the Ph.D. degree from National University of Defense Technology in 2004. He is now a associate professor at National University of Defense Technology. His research interest covers mission planning , scheduling and articial intelligence.