

Research Article

Approximate Path Searching Method for Single-Satellite Observation and Transmission Task Planning Problem

Shuang Peng, Hao Chen, Jun Li, and Ning Jing

School of Electronic Science and Engineering, National University of Defense Technology, Hunan, Changsha 410073, China

Correspondence should be addressed to Hao Chen; hchen@nudt.edu.cn

Received 1 October 2017; Revised 19 November 2017; Accepted 27 November 2017; Published 27 December 2017

Academic Editor: Simone Cinquemani

Copyright © 2017 Shuang Peng 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.

Satellite task planning not only plans the observation tasks to collect images of the earth surface, but also schedules the transmission tasks to download images to the ground station for users' using, which plays an important role in improving the efficiency of the satellite observation system. However, most of the work to our knowledge, scheduling the observation and transmission tasks separately, ignores the correlation between them in resource (e.g., energy and memory) consumption and acquisition. In this paper, we study the single-satellite observation and transmission task planning problem under a more accurate resource usage model. Two preprocessing strategies including graph partition and nondominated subpaths selection are used to decompose the problem, and an improved label-setting algorithm with the lower bound cutting strategy is proposed to maximize the total benefit. Finally, we compare the proposed method with other three algorithms based on three data sets, and the experimental result shows that our method can find the near-optimal solution in much less time.

1. Introduction

Earth observation satellite (EOS) collects images of the earth surface and downloads them to the ground station for users' using, which has been widely used in environmental monitoring, resource management, agricultural analysis, military reconnaissance, and another aspect of life [1–3]. At present, the contradiction between the overloaded requirements for remote sensing images and the limited EOS resources is increasingly acute. Satellite task planning is to optimize the observation and transmission activities to maximize the total benefit that receives much attention from the government and space agency [1, 4–6].

As is shown in Figure 1, the EOS orbits the earth. The observation payload takes images and stores them in the storage when a target is in the view of the EOS; this process is called “observation.” The transmission payload downloads images to the ground and frees the corresponding storage when the EOS is in the visible range of a ground station; this process is called “transmission.” Energy is consumed in the process of observation and transmission and acquired when the EOS is in the sun. There is a strong correlation between the observation tasks and the transmission tasks in resource

(e.g., energy and memory) consumption and acquisition, which makes the problem intractable to solve.

In this paper, we study the single-satellite observation and transmission tasks planning problem (SOTTP) under a more accurate resource usage model. The main contribution includes the following: (1) we develop a directed acyclic graph model and propose two preprocessing strategies including graph partition and nondominated subpaths selection. These two preprocessing strategies decompose the graph model into a series of subgraphs that effectively reduce the complexity of the problem. (2) We construct an equivalent mathematical model and propose an improved label-setting algorithm with the lower bound cutting strategy to solve the SOTTP. This strategy removes the low-benefit paths in each subgraph that improves the search efficiency. (3) Numerical experiments on three data sets show that our method can find the near-optimal solution in much less time.

This paper is organized as follows. Section 2 places the related work of our research. Section 3 describes the problem which includes the time-dependent resource change relation and the resource constraints. In Section 4, two preprocessing strategies including graph partition and nondominated subpaths selection are used to decompose the problem, and

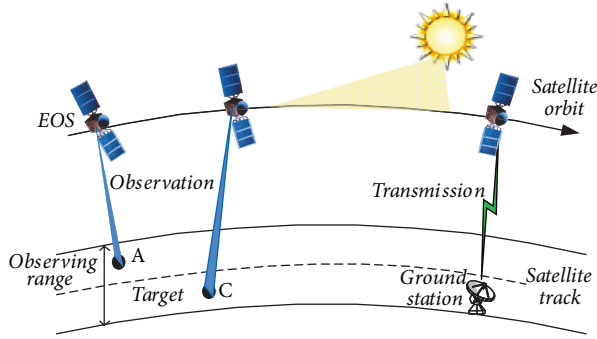


FIGURE 1: The operation of an EOS.

an equivalent mathematical model is developed. In Section 5, an improved label-setting algorithm with the lower bound cutting strategy is adopted to solve the problem. Section 6 describes experimental result on three data sets. Section 7 provides concluding remarks and future works.

2. Relate Works

Satellite task planning is a typical combinatorial optimization problem and proved to be NP-Hard [1, 2, 6, 7], which has been widely concerned by scholars and made great achievements. According to the planning approach, the research can be mainly classified into three categories: separate planning, hierarchical planning, and coordinate planning.

In separate planning, the observation tasks and the transmission tasks are scheduled separately. Either the satellite observation task planning problem (SOTP) or the satellite transmission task planning problem (STTP, it is also called the satellite ground station scheduling problem) is generally formulated by some mathematical models, such as graph model [8–10], integer programming model [1, 2, 7, 8, 11, 12], or constraint satisfaction programming model [6, 13, 14]. Then they are solved by metaheuristic algorithms (like genetic algorithm [3, 6, 13], tabu search [14, 15], and ant colony optimization [9]), mathematical programming methods (like Lagrangian relaxation [2, 12], branch and cut [7], and column generation [8]), and other domain knowledge based heuristic approaches [10, 16]. However, the memory constraint is ignored or the storage is assumed to be large enough in the SOTP [1–3, 8–11, 14–18]. The energy and memory constraints are neglected or simplified, and the minimum download time is used as the primary optimization objective in the STTP [6, 7, 12, 13, 19–21]. Though the separate planning approach simplifies the problem, it disregards the correlation between the observation tasks and the transmission tasks in resource (e.g., energy and memory) consumption and acquisition.

In hierarchical planning [22, 23], the observation tasks are first planned by an algorithm; then the transmission tasks are scheduled by another algorithm based on the selected observation tasks in the previous stage. The quality of solution not only depends on the performance of the two algorithms, but also relates to the way to bond them. The planning

result of the observation tasks can be easily destroyed while scheduling the transmission tasks.

In coordinate planning [5, 24, 25], the observation tasks and the transmission tasks are scheduled together. Such studies are not much and still in the exploring stage. Li et al. [24] proposed a K -shortest path genetic algorithm to balance the response time and resource utilization. Wang et al. [25] studied the scheduling problem of four optical satellites and four SAR satellites and presented a priority-based heuristic method to maximize the rewards of the images taken and transmitted. Chen et al. [5] considered the specific constraints of electromagnetic detection satellites and adopted a genetic algorithm with solution repairing strategy to solve the problem. Because of the limitation of the TT&C (Telemetry, Tracking, and Command) network resource, the maximum working time is used to model the resource constraint (e.g., the working time of the EOS in each orbital period is the same) in the above research. In the real world, the resource (e.g., energy and memory) status is continuous and correlated [7, 26, 27]. The more resources consumed in the current orbital period, the fewer resources available in the next cycle. It is evident that the traditional resource usage model is imprecise that usually limits the capacity of the EOS. With the emergence of autonomous satellite [26] and the use of data relay satellite [28], it is possible to get the resource status in real time and use the onboard resource usage model to accurately represent the realistic constraints of the EOS rather than the traditional model.

In this paper, we study the single-satellite observation and transmission task planning problem from the perspective of resource consumption and acquisition and coordinately plan the observation tasks and the transmission tasks. A more accurate resource usage model is considered, and an approximate path searching method is proposed to solve the problem.

3. Problem Description

In the problem, we consider the following hypotheses:

- (1) The observation tasks and the transmission tasks are given before planning.
- (2) The energy and the memory are renewable resources, but they are limited by the battery capacity and storage capacity, respectively.
- (3) The EOS may view more than one ground station simultaneously, but it can only download the images to one ground station at a time.
- (4) The EOS cannot transmit the images to the ground station during observing.
- (5) Once an observation task or a transmission task is executed, it cannot be interrupted or removed.

3.1. Planning Horizon. $H = [T_a, T_e]$ is the planning horizon, where T_a is the start time and T_e is the end time. All observation and transmission tasks must be in the planning horizon.

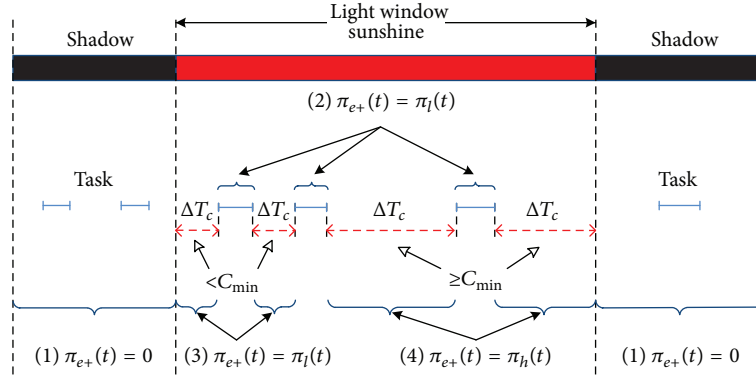


FIGURE 2: The charging rate in different situations.

3.2. Data Set

(1) *Task Set ST*. The observation task and the transmission task are collectively referred to as task. $\forall i \in ST$, it is expressed by a 3-tuple (ta_i, te_i, b_i) :

- (i) ta_i : the start time of task i , which is a constant
- (ii) te_i : the end time of task i , which is a constant
- (iii) b_i : the benefit of task i . For an observation task, $b_i \in [1, 40]$, the higher the value, the more important the task. For a transmission task, it does not take any pictures except downloading them to the ground station such that the benefit is set as zero.

(2) *Light Window Set LW*. The EOS can only recharge the battery in the light window. $\forall k \in LW$, it is expressed by a 2-tuple (la_k, le_k) :

- (i) la_k : (a constant) the start time of light window k , which means the EOS goes into the sunshine from the shadow
- (ii) le_k : (a constant) the end time of light window k , which means the EOS goes into the shadow from the sunshine.

3.3. *Temporal Relation between Tasks*. The EOS cannot perform two tasks at a time in our problem. As a result, a setup time $Ts(i, j)$ ($Ts(i, j) \leq Ts_m$, where Ts_m is the maximum setup

time) is required for any two tasks i and j , $\forall i, j \in ST$. Note that $Ts(i, j)$ is a constant and predefined before planning. Then the temporal relation can be presented as follows:

$$\begin{aligned}
 ta_i < ta_j &\longrightarrow \\
 ta_j - te_i &\geq Ts(i, j), \\
 \forall i, j &\in ST.
 \end{aligned} \tag{1}$$

3.4. Time-Dependent Resource Change Relation

(1) *The Change of Energy*. The EOS collects energy by the solar panel and stores it in the battery to support the basic functionality and perform tasks. The change of energy is described as follows:

$$\begin{aligned}
 \Delta e(t_1, t_2) &= \int_{t_1}^{t_2} (\pi_{e+}(t) - \pi_{e-}(t) - \phi_{e-}(t)) dt, \\
 t_2 &> t_1,
 \end{aligned} \tag{2}$$

where $\pi_{e-}(t)$ is the energy consumption rate of basic functionality, $\phi_{e-}(t)$ is the energy consumption rate of satellite payloads, and $\pi_{e+}(t)$ is the charging rate. If the EOS wants to charge in the high charging rate, it must adjust the attitude that points the solar panel to the sun when it is idle (e.g., no task to do) and a minimum charging time is needed. As a result, $\pi_{e+}(t)$ is defined by a piecewise function (see (3)), as is shown in Figure 2.

$$\pi_{e+}(t) = \begin{cases} 0 & (1) \text{ if } \forall k \in LW \text{ such that } t \notin [la_k, le_k] \\ \pi_l(t) & (2) \text{ if } \exists k \in LW, \exists i \in ST \text{ such that } t \in [la_k, le_k], t \in [ta_i, te_i] \\ \pi_l(t) & (3) \text{ if } \exists k \in LW, \forall i \in ST \text{ such that } t \in [la_k, le_k], t \notin [ta_i, te_i], \Delta T_c(k, t) < C_{\min} \\ \pi_h(t) & (4) \text{ if } \exists k \in LW, \forall i \in ST \text{ such that } t \in [la_k, le_k], t \notin [ta_i, te_i], \Delta T_c(k, t) \geq C_{\min}, \end{cases} \tag{3}$$

where $\pi_l(t)$ is the low charging rate, $\pi_h(t)$ is the high charging rate, C_{\min} ($C_{\min} > Ts_m$) is the minimum charging time, and $\Delta T_c(k, t)$ is the available charging time which is defined by

$$\begin{aligned}
 \Delta T_c(k, t) &= \min \{le_k, ta_i \mid i \in ST, ta_i > t\} \\
 &\quad - \max \{la_k, ta_i \mid i \in ST, ta_i < t\}.
 \end{aligned} \tag{4}$$

(2) *The Change of Data.* The change of data is described as

$$\Delta d(t_1, t_2) = \int_{t_1}^{t_2} \phi_d(t) dt, \quad t_2 > t_1, \quad (5)$$

where $\phi_d(t)$ is the change rate of data, $\phi_d(t) > 0$ when the EOS is performing an observation task at time t , and $\phi_d(t) < 0$ when the EOS is performing a transmission task at time t ; otherwise $\phi_d(t) = 0$.

3.5. Energy and Memory Constraints. The energy level cannot exceed the lower bound and upper bound of energy at any time because of the limitation of battery capacity. When the energy level reaches the upper bound, it will stop increasing. Let $e(t)$ be the energy level at time t , E_{\min} be the lower bound, and E_{\max} be the upper bound; then the energy constraint is represented as follows:

$$e(t) = \min \{E_{\max}, e(t_1) + \Delta e(t_1, t)\} \geq E_{\min}, \quad t > t_1. \quad (6)$$

The memory constraint is similar to the energy constraint. The difference is that the data level will stop decreasing when it reaches the lower bound of memory (e.g., all images are downloaded to the ground). Let $d(t)$ be the data level at time t and D_{\min} ($D_{\min} = 0$ in the problem) and D_{\max} be the lower bound and upper bound of memory; then the memory constraint is formulated as follows:

$$d(t) = \max \{D_{\min}, d(t_1) + \Delta d(t_1, t)\} \leq D_{\max}, \quad t > t_1. \quad (7)$$

Definition 1 (resource vector). Let p be the plan in the period $[t_1, t_2]$; then the resource usage of plan p can be expressed by the resource vector $R = (\Delta e_{\max}, \Delta e_{\min}, \Delta d_{\max}, \Delta d_{\min}, \Delta e_{\text{end}}, \Delta d_{\text{end}})$ (Figure 3); for the computation, one can refer to Appendix A.

- (i) $\Delta e_{\max} \geq 0$ and $\Delta e_{\min} \leq 0$: the maximum and minimum change of energy while performing plan p .
- (ii) $\Delta d_{\max} \geq 0$ and $\Delta d_{\min} \leq 0$: the maximum and minimum change of data while performing plan p .
- (iii) Δe_{end} and Δd_{end} : the change of energy and memory after plan p is performed.

Theorem 2. Let $e(t_1)$ and $d(t_1)$ be the energy and data level at time t_1 . If p is a feasible plan in the period $[t_1, t_2]$, it must satisfy the following constraints:

$$\begin{aligned} \max \{ \Delta e(t_1, t'), t' \in [t_1, t] \} - \Delta e(t_1, t) \\ \leq E_{\max} - E_{\min}, \quad t \in [t_1, t_2], \end{aligned} \quad (8a)$$

$$\begin{aligned} \Delta d(t_1, t) - \min \{ \Delta d(t_1, t'), t' \in [t_1, t] \} \\ \leq D_{\max} - D_{\min}, \quad t \in [t_1, t_2], \end{aligned} \quad (8b)$$

$$e(t_1) + \Delta e_{\min} \geq E_{\min}, \quad (8c)$$

$$d(t_1) + \Delta d_{\max} \leq D_{\max}. \quad (8d)$$

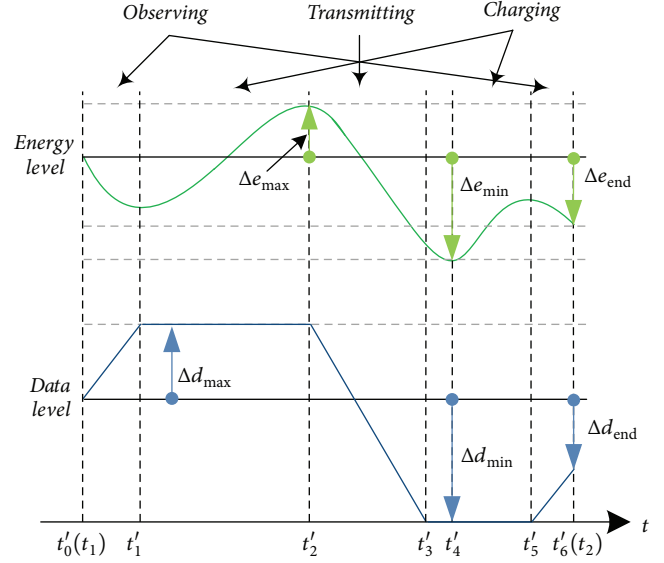


FIGURE 3: The resource vector in the period $[t_1, t_2]$.

Proof. See Appendix B. \square

Theorem 3. Let $e(t_1)$ and $d(t_1)$ be the energy and data level at time t_1 . If p is a feasible plan in the period $[t_1, t_2]$, the energy and data level at time t_2 can be computed as follows:

$$e(t_2) = \min \{E_{\max} - \Delta e_{\max}, e(t_1)\} + \Delta e_{\text{end}}, \quad (9a)$$

$$d(t_2) = \max \{D_{\min} - \Delta d_{\min}, d(t_1)\} + \Delta d_{\text{end}}. \quad (9b)$$

Proof. See Appendix C. \square

3.6. Optimization Objective. The objective of the SOTTP is to plan the observation and transmission tasks in the planning horizon which maximize the total benefit without violating all constraints.

4. The Preprocessing Strategies and the Planning Model

In this section, we first develop a directed acyclic graph model for the problem. Then the graph is partitioned into several subgraphs by the graph partition strategy, and the nondominated subpaths in each subgraph are searched by the nondominated subpaths selection strategy. Finally, an equivalent planning model is established.

4.1. The Directed Acyclic Graph Model. Given the tasks and sorting them in chronological order, the directed acyclic graph $G = (V, E)$ (Figure 4) is used to model the problem, where V is the set of nodes and E is the set of edges. Each node $v_i \in V$ is expressed as (ta_i, te_i) , where ta_i, te_i are the start time and end time of task i , $i \in \text{ST}$. Each edge $(i, j) \in E$ is denoted by (R_{ij}, b_{ij}) , where $R_{ij} = (\Delta e_{\max}^{ij}, \Delta e_{\min}^{ij}, \Delta e_{\text{end}}^{ij}, \Delta d_{\max}^{ij}, \Delta d_{\min}^{ij}, \Delta d_{\text{end}}^{ij})$ is the resource vector (edge (i, j) is a plan in the period $[te_i, te_j]$, see Definition 1) and b_{ij} is equal to the

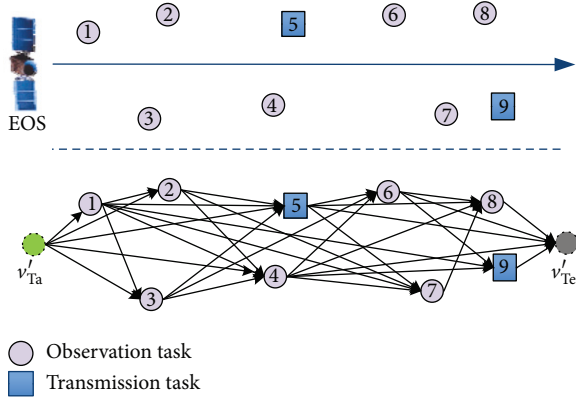


FIGURE 4: The directed acyclic graph model.

benefit of task j , $j \in ST$. $\forall v_i, v_j \in V$, if they satisfy the constraints (1), (8a), and (8b), there is a directed edge point from v_i to v_j . There are also two virtual nodes $v'_{Ta} = (Ta, Ta)$ (the source node) and $v'_{Te} = (Te, Te)$ (the sink node) in the graph. The goal is to find an elementary path from v'_{Ta} to v'_{Te} that maximizes the total benefit without violating the energy and memory constraints.

Some notations used in the model are defined as follows:

- (i) E_o, D_o : the amounts of energy and data that the EOS stored at the beginning of the planning horizon
- (ii) E_{\min}, E_{\max} : the minimum and maximum allowable amounts of energy to be stored in the battery (e.g., the lower bound and upper bound of energy)
- (iii) D_{\min}, D_{\max} : the minimum and maximum allowable amounts of data to be stored in the storage (e.g., the lower bound and upper bound of memory)
- (iv) $x_{ij} \in \{0, 1\}$: the decision variable; $x_{ij} = 1$ means edge (i, j) is selected; otherwise it is not, $1 \leq i < j \leq n$
- (v) $y_j \geq 0$: the decision variable, which represents the total amount of energy spilled throughout edges that point to node v_j , $1 \leq j \leq n$
- (vi) $z_j \geq 0$: the decision variable, which represents the total amount of data spilled throughout edges that point to node v_j , $1 \leq j \leq n$.

Then the mathematical model can be formulated as follows:

$$\text{Maximize } \sum_{i=1}^n \sum_{j=1}^n b_{ij} * x_{ij} \quad (10)$$

$$\text{Subject to } \sum_{j=1}^n x_{1j} = 1 \quad (11)$$

$$\sum_{i=1}^n x_{in} = 1 \quad (12)$$

$$\sum_{i=1}^{j-1} x_{ij} \leq 1, \quad 1 < j < n \quad (13)$$

$$\sum_{i=1}^{j-1} x_{ij} = \sum_{k=i+1}^n x_{jk}, \quad 1 < j < n \quad (14)$$

$$E_o + \sum_{j=2}^{k-1} \left(\sum_{i=1}^{j-1} \Delta e_{\text{end}}^{ij} * x_{ij} - y_j \right) + \sum_{i=1}^{k-1} \Delta e_{\text{max}}^{ik} * x_{ik} - y_k \leq E_{\text{max}}, \quad 1 < k \leq n \quad (15)$$

$$E_o + \sum_{j=2}^{k-1} \left(\sum_{i=1}^{j-1} \Delta e_{\text{end}}^{ij} * x_{ij} - y_j \right) + \sum_{i=1}^{k-1} \Delta e_{\text{min}}^{ik} * x_{ik} \geq E_{\text{min}}, \quad 1 < k \leq n \quad (16)$$

$$E_o + \sum_{j=2}^{k-1} \left(\sum_{i=1}^{j-1} \Delta d_{\text{end}}^{ij} * x_{ij} + z_j \right) + \sum_{i=1}^{k-1} \Delta d_{\text{max}}^{ik} * x_{ik} \leq D_{\text{max}}, \quad 1 < k \leq n \quad (17)$$

$$E_o + \sum_{j=2}^{k-1} \left(\sum_{i=1}^{j-1} \Delta d_{\text{end}}^{ij} * x_{ij} + z_j \right) + \sum_{i=1}^{k-1} \Delta d_{\text{min}}^{ik} * x_{ik} + z_k \geq D_{\text{min}}, \quad 1 < k \leq n. \quad (18)$$

The objective function (10) is used to maximize the total benefit over the planning horizon. Constraints (11), (12), (13), and (14) are the outdegree and indegree constraints of each node. Constraints (15), (16), (17), and (18) ensure that the energy and data level at each node cannot exceed the lower bound and upper bound of energy and memory.

4.2. Graph Partition. In this subsection, a virtual common node based edge-cut method is used to partition the graph. The definition of the virtual common node is as follows.

Definition 4. Partition point and virtual common node: given a time t_A , let $V(L_A)$ be the set of nodes whose start time is less than t_A and $V(R_A)$ be the set of nodes whose end time is greater than t_A . If the following two conditions are satisfied (as is shown in Figure 5), then time t_A is a partition point and node $v'_A = (t_A, t_A)$ is a virtual common node of graph G .

$$(1) \forall k \in LW \rightarrow t_A \notin [la_k, le_k], \min\{ta_j \mid v_j \in V(R_A)\} - \max\{te_i \mid v_i \in V(L_A)\} \geq Ts_m.$$

$$(2) \exists k \in LW \rightarrow t_A - \max\{la_k, te_i \mid v_i \in V(L_A)\} \geq C_{\min}, \min\{le_k, ta_j \mid v_j \in V(R_A)\} - t_A \geq C_{\min}.$$

If node $v'_A = (t_A, t_A)$ is a virtual common node of graph G , then the edges that connect the nodes on the left and right

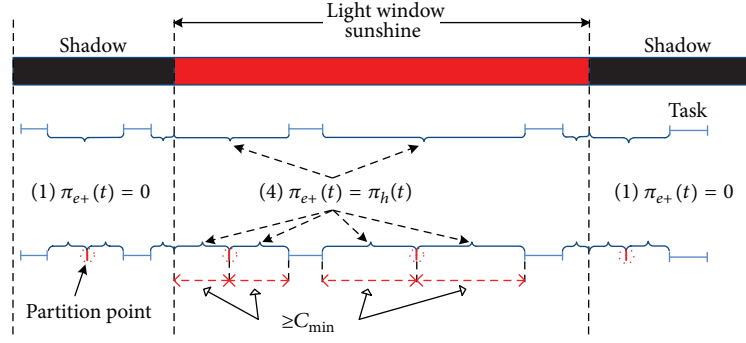


FIGURE 5: The partition point in a plan.

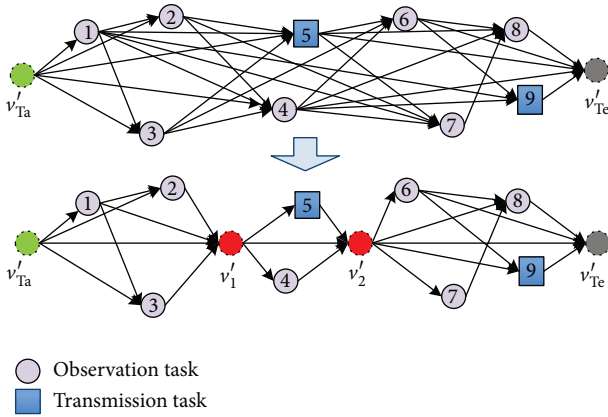


FIGURE 6: The DAG model after graph partition.

side of v'_A are updated (e.g., edges (i, A) , (A, j) are added in the graph and edge (i, j) is removed, $\forall v_i \in V(L_A), \forall v_j \in V(R_A)$). After all virtual common nodes are found and the edges are updated, the graph is partitioned (as is shown in Figure 6). Note that there is also an edge that connects two neighbor virtual common nodes.

Lemma 5. *If node $v'_A = (t_A, t_A)$ is a virtual common node of graph G , the connectivity and the resource vector between any two nodes are consistent before and after graph partition.*

Proof. From requirement (1) and requirement (2), $\forall v_i \in V(L_A), \forall v_j \in V(R_A)$, $ta_j - te_i > \min\{Ts_m, C_{min}\} = Ts_m \geq Ts(i, j)$, so there is an edge or path connecting node v_i and node v_j before and after graph partition. That is, the connectivity is consistent.

$\forall v_i \in V(L_A), \forall v_j \in V(R_A)$, let $\pi'_{e^+}(t)$, $\pi''_{e^+}(t)$, and $\pi'''_{e^+}(t)$ be the charging rate on edges (i, j) , (i, A) , and (A, j) , respectively. If requirement (1) or requirement (2) is met, from formula (3), we know that $\pi'_{e^+}(t_A) = \pi''_{e^+}(t_A) = \pi'''_{e^+}(t_A)$; $\forall t \in [te_i, t_A]$, $\pi'_{e^+}(t) = \pi''_{e^+}(t)$; $\forall t \in [t_A, ta_j]$, $\pi'_{e^+}(t) = \pi'''_{e^+}(t)$. That is, the charging rates in the period $[te_i, ta_j]$ are the same before and after graph partition. From Definition 1, we can conclude that the resource vector is consistent. \square

By using the graph partition strategy, the graph model can be described as $G = \{G_1, G_2, \dots, G_s\}$, where G_i is the

i th subgraph and s is the number of subgraphs. Let $V' = \{v'_0, \dots, v'_s\}$ be the set of virtual nodes where $v'_0 = v'_{Ta}$ and $v'_s = v'_{Te}$; $V(G_i)$ is the set of nodes in G_i ; then $V(G_i) \cap V(G_{i+1}) = v'_i$, $\bigcup_{1 \leq i \leq s} V(G_i) = V(G) \cup V'$.

4.3. Nondominated Subpaths Selection. After the graph is partitioned, the label-setting algorithm [18, 29, 30] is used to search the nondominated subpaths in each subgraph. The dominance relation used in this algorithm is relaxed, which is defined as follows.

Definition 6 (dominance relation). Let $P_{i,j}$ be the set of paths from node v_i to node v_j . $\forall p_m, p_n \in P_{i,j}$, if they satisfy one of the following three conditions, then path p_m is dominant to path p_n and expressed as $p_m \geq p_n$, where b_m is the total benefit of p_m , Δe_{end}^m and Δd_{end}^m are the change of energy and memory of p_m (see Definition 1), and α, β, γ are the relaxation factor of benefit, energy, and memory, respectively. In these conditions, $b_m, \alpha, \beta, \gamma$ are nonnegative, and $\Delta e_{end}^m, \Delta d_{end}^m$ can be positive or negative.

- (1) $b_m * (1 + \alpha) \geq b_n$, $\Delta e_{end}^m \geq \Delta e_{end}^n$, and $\Delta d_{end}^m \leq \Delta d_{end}^n$.
- (2) $b_m \geq b_n$, $\Delta e_{end}^m * (1 + \beta * (\Delta e_{end}^m / |\Delta e_{end}^m|)) \geq \Delta e_{end}^n$, and $\Delta d_{end}^m \leq \Delta d_{end}^n$.
- (3) $b_m \geq b_n$, $\Delta e_{end}^m \geq \Delta e_{end}^n$, and $\Delta d_{end}^m * (1 - \gamma * (\Delta d_{end}^m / |\Delta d_{end}^m|)) \leq \Delta d_{end}^n$.

For $p_n \in P_{i,j}$, if $\nexists p_m \in P_{i,j}$ with $m \neq n$ that satisfies $p_m \geq p_n$, then p_n is a nondominated path; otherwise it is a dominated path.

The process of nondominated subpaths selection in subgraph G_i is to search the nondominated paths from node v'_{i-1} to node v'_i ; for details, one can refer to [30]. Note that the subpaths violating constraint (8a) or (8b) are also removed in the process.

4.4. The Equivalent Planning Model. Through preprocessing, the graph model is partitioned into several subgraphs, and the problem is transformed to search a feasible subpath from each subgraph without violating the resource constraint. An equivalent planning model is developed in this subsection. The notations used in the model are summarised as follows.

Parameters

- (i) s : the number of subgraphs in graph G
- (ii) n_i : the number of subpaths in subgraph G_i , $1 \leq i \leq s$
- (iii) tg_i : the end time of subgraph G_i , which is equal to the end time of node v'_i
- (iv) P_i : the set of subpaths (from node v'_{i-1} to node v'_i) in subgraph G_i
- (v) Q_i : the set of paths (from node v'_{Ta} to node v'_i) in subgraph G_i
- (vi) p_{ij} : the j th subpath in P_i
- (vii) b'_{ij} : the total benefit of subpath p_{ij}
- (viii) $R'_{ij} = (\Delta e'_{max}, \Delta e'_{min}, \Delta d'_{max}, \Delta d'_{min}, \Delta e'_{end}, \Delta d'_{end})$: the resource factor of subpath p_{ij} (a plan in the period $[tg_{i-1}, tg_i]$); for details, refer to Definition 1
- (ix) $E_o, D_o, E_{min}, E_{max}, D_{min}, D_{max}$: refer to Section 4.1.

Variables

- (i) $x'_{ij} \in \{0, 1\}$: the binary value that represents whether subpath p_{ij} is selected, $1 \leq i \leq s, 1 \leq j \leq n_i$
- (ii) $e(tg_i)$: the energy level at the end of subgraph G_i , $1 \leq i \leq s$
- (iii) $d(tg_i)$: the data level at the end of subgraph G_i , $1 \leq i \leq s$.

Then the planning model is formulated as follows:

$$\text{Maximize } \sum_{i=1}^s \sum_{j=1}^{n_i} b'_{ij} * x'_{ij} \quad (19)$$

$$\text{Subject to } \sum_{j=1}^{n_i} x'_{ij} = 1, \quad 1 \leq i \leq s \quad (20)$$

$$e(tg_0) = E_o \quad (21)$$

$$d(tg_0) = D_o \quad (22)$$

$$\begin{aligned} & e(tg_i) \\ &= \min \left\{ E_{\max} - \sum_{j=1}^{n_i} \Delta e'_{\max} * x'_{ij}, e(tg_{i-1}) \right\} \\ & \quad + \sum_{j=1}^{n_i} \Delta e'_{\text{end}} * x'_{ij}, \quad 1 \leq i \leq s \end{aligned} \quad (23)$$

$$\begin{aligned} & d(tg_i) \\ &= \max \left\{ D_{\min} - \sum_{j=1}^{n_i} \Delta d'_{\min} * x'_{ij}, d(tg_{i-1}) \right\} \end{aligned} \quad (24)$$

$$\begin{aligned} & \quad + \sum_{j=1}^{n_i} \Delta d'_{\text{end}} * x'_{ij}, \quad 1 \leq i \leq s \\ & e(tg_{i-1}) + \sum_{j=1}^{n_i} \Delta e'_{\min} * x'_{ij} \geq E_{\min}, \quad 1 \leq i \leq s \end{aligned} \quad (25)$$

$$\begin{aligned} & d(tg_{i-1}) + \sum_{j=1}^{n_i} \Delta d'_{\max} * x'_{ij} \leq D_{\max}, \\ & \quad 1 \leq i \leq s. \end{aligned} \quad (26)$$

The objective function (19) is the same as (10). Equation (20) enforces that there is one and only one subpath that can be selected in each subgraph. Constraints (21) and (22) initialize the amount of energy and data that satellite stored at the beginning of the planning horizon. Constraint (23) defines the energy level at the end of each subgraph. Constraint (24) defines the data level at the end of each subgraph. Constraint (25) ensures that the energy level cannot be less than the lower bound of energy in each subgraph. Constraint (26) ensures that the data level cannot be greater than the upper bound of memory in each subgraph.

5. The Improved Label-Setting Algorithm

In this section, we propose an improved label-setting algorithm with the lower bound cutting strategy (LSLBC) to search the near-optimal solution of the SOTTP. The label-setting algorithm (LS) is an exact path searching algorithm that can find all nondominated paths in each subgraph. However, only the nondominated path with the maximum benefit is selected in our problem, and low-benefit paths in each subgraph are mostly useless. The lower bound cutting strategy is used to remove the paths in each subgraph whose benefit is less than the lower bound, which avoids enumerating all nondominated paths and improves search efficiency.

5.1. The Lower Bound Cutting Strategy. The lower bound of each subgraph is the key of the lower bound cutting strategy, but they are hard to be computed. In fact, there is a connection between them and the lower bound of the problem, such that we can compute the approximate lower bound of a subgraph based on this relation.

5.1.1. The Lower Bound of the Problem. We design a two-phase hierarchical algorithm (called GS-BC) to find the lower bound of the problem. Before discussing the algorithm, the concept it used is given as follows.

```

Input: graph  $G = \{G_1, \dots, G_s\}$ .
Output: solution  $P^* = \{p_1^*, \dots, p_s^*\}$  /*  $p_i^*$  is the selected subpath in  $G_i$ . */
(1)  $\forall i \in [1, s], p_i^* \leftarrow \text{null}$ ;
(2) for all  $G_i \in G$  do
(3)    $p_j \leftarrow \text{maxbenefit}(P_i)$ ; /*  $p_j$  is the subpath with the maximum benefit in  $P_i$ . */
(4)    $p_i^* \leftarrow p_j$ ;
(5)    $P_i \leftarrow P_i \setminus p_j$ ;
(6)   if  $\text{isconflict}(P^*)$  then /* whether the energy constraint is violated. */.
(7)     while  $\text{isconflict}(P^*)$  do
(8)        $m \leftarrow \text{repairid}(P^*)$ ; /*  $m$  is the index of the subpath with the minimum energy efficiency in  $P^*$ . */
(9)        $p_k \leftarrow \text{maxbenefit}(P_m)$ ; /*  $p_k$  is the subpath with the maximum benefit in  $P_m$ . */
(10)       $p_m^* \leftarrow p_k$ ;
(11)       $P_m \leftarrow P_m \setminus p_k$ ;
(12)    end while
(13)  end if
(14) end for

```

ALGORITHM 1: Energy efficiency based greedy search algorithm.

Definition 7 (energy efficiency).

$$\eta(p_{ij}) = \frac{\Delta e_{\text{end}}^{ij'} - \max \left\{ \Delta e_{\text{end}}^{ik'} \mid p_{ik} \in P_i \right\}}{b'_{ij} + \varepsilon}, \quad (27)$$

where $\eta(p_{ij})$ is the energy efficiency of subpath p_{ij} and ε is the smallest constant in the computer used to avoid zero-division error.

The energy efficiency represents the amount of energy consumed per unit of benefit and satisfies $\eta(p_{ij}) \leq 0$.

In the first phase of GS-BC, the observation tasks are planned by the energy efficiency based greedy search algorithm (GS, as is shown in Algorithm 1). This algorithm selects a subpath from each subgraph one by one until reaching the end. When it searches a subgraph, the subpath with the maximum benefit is added to the solution. If the energy constraint is violated, then the subpath with the minimum energy efficiency in the solution is replaced by another subpath (it is not searched and has the maximum benefit) until the conflict is removed. In the second phase of GS-BC, the branch and cut algorithm (BC) [7] is used to schedule the transmission tasks based on the selected observation tasks in the first phase.

5.1.2. The Lower Bound of the Subgraph. Let $P^* = \{p_1^*, p_2^*, \dots, p_s^*\}$ be the solution of GS-BC, p_i^* be the selected subpath in subgraph G_i , and b_i^* be the benefit of subpath p_i^* ; then we define the lower bound of a subgraph as

$$L(k) = f_1 \left(\frac{\sum_{i=1}^k b_i^*}{\sum_{i=1}^s b_i^*} \right) * \sum_{i=1}^k b_i^*, \quad 1 \leq k \leq s, \quad (28)$$

where $f_1(x)$ is the discount function that represents the relation between the lower bound of a subgraph and the lower bound of the problem. $f_1(x)$ is obtained by the statistical method; for details, refer to Section 6.2.2.

5.2. The Improved Label-Setting Algorithm. After the lower bound of each subgraph is computed, the improved label-setting algorithm with the lower bound cutting strategy is used to find the near-optimal solution of the problem. This algorithm incrementally searches the nondominated paths from the first subgraph to the last subgraph one by one until reaching the end. When it searches a subgraph, the low-benefit paths are removed by the lower bound cutting strategy, and the dominated paths are discarded based on the dominance relation (see Definition 6) that avoids exploring the whole paths in the solution space. For details, refer to Algorithm 2.

6. Computational Experiments

6.1. Test Data. In the experiment, there are one satellite and four ground stations [25] (Kashi, Miyun, Sanya, and Beiji). The EOS is selected from the STK (Satellite Tool Kit) satellite database. The planning horizon is set as a typical 1 day from 2017-04-01 00:00:00 to 2017-04-02 00:00:00.

There is no acknowledged benchmark data set in the single-satellite observation and transmission tasks planning problem, so we randomly generate three data sets: regional city data set, global city data set, and global point data set. Cities in the regional city data set and global city data set are selected from the STK city database. In the regional city data set, all targets are randomly distributed in the area: $0^\circ\text{N} \sim 65^\circ\text{N}$ and $0^\circ\text{E} \sim 150^\circ\text{E}$. In the global city data set and global point data set, all targets are randomly distributed in the area: $-65^\circ\text{S} \sim 65^\circ\text{N}$ and $-180^\circ\text{W} \sim 180^\circ\text{E}$. The access times between the EOS and the ground station or the EOS and the target are calculated by STK. Each access time is corresponding to a task.

There are nine problem sets in each data set. Each problem set contains ten problems (i.e., a total of 90 problems in each data set). The number of observation tasks in each problem set is from 100 to 500 with step of 50.


```

Input: graph  $G = \{G_1, \dots, G_s\}$ .
Output: the optimal path  $q_s^*$ 
(1) for  $i = 2, \dots, s$  do
(2)   for all  $p_i \in P_i$  do /*  $P_i$  is set of subpaths in subgraph  $G_i$ . */
(3)     for all  $q_{i-1} \in Q_{i-1}$  do /*  $Q_{i-1}$  is set of paths in subgraph  $G_{i-1}$ . */
(4)       if  $\text{notconflict}(q_{i-1}, p_i)$  then /*  $p_i$  must satisfy the constraint when adding it to  $q_{i-1}$ . */
(5)          $q_i \leftarrow \text{append}(q_{i-1}, p_i)$ ; /* add  $p_i$  to  $q_{i-1}$ . */
(6)          $Q_i \leftarrow Q_i \cup q_i$ ;
(7)       end if
(8)     end for
(9)    $L_i \leftarrow \text{getlowerbound}()$ ; /*  $L_i$  is the lower bound of subgraph  $G_i$ . */
(10)  for  $j = 1, \dots, \text{num}(Q_i)$  do /*  $\text{num}(Q_i)$  is the number of paths in  $Q_i$ . */
(11)    if  $b_j < L(i)$  then /* the total benefit must greater than the lower bound. */
(12)       $Q_i \leftarrow Q_i \setminus q_j$ ;
(13)    end if
(14)  end for
(15)  for  $j = 1, \dots, \text{num}(Q_i)$  do
(16)    for  $k = j + 1, \dots, \text{num}(Q_i)$  do
(17)      if  $q_j \leq q_k$  then
(18)         $Q_i \leftarrow Q_i \setminus q_j$ ;
(19)        break;
(20)      else if  $q_j \geq q_k$  then
(21)         $Q_i \leftarrow Q_i \setminus q_k$ ;
(22)      end if
(23)    end for
(24)  end for
(25) end for
(26) end for
(27)  $q_s^* \leftarrow \text{maxbenefit}(Q_s)$ ; /*  $q_s^*$  is the path with the maximum benefit in  $Q_s$ . */

```

ALGORITHM 2: Improved label-setting algorithm.

All algorithms in the experiment are implemented and compiled with Microsoft C# compiler and run on Windows 7 OS with an Intel E5-2650 2.30 GHz CPU and 8 GB memory.

6.2. Parameter Study

6.2.1. The Effect of the Relaxation Factors. To investigate the effect of the three relaxation factors in our method, we conduct several experiments based on the global city data set.

Figures 7 and 8 show the result of LS under single relaxation factor. The horizontal axis represents the number of observation tasks, the vertical axis represents the average benefit or average CPU time, and the legend " $\alpha = 0.001$ " means the relaxation factor of benefit is 0.001 and the other two relaxation factors are set as zero.

In Figure 7, the average benefit falls with the increase of each relaxation factor. In Figure 7(a), the average benefit is drastically reduced when α is greater than 0.002. It reveals that LS is sensitive to the relaxation factor of benefit. In Figures 7(b) and 7(c), the gap of benefit is pretty narrow when β or γ is less than 0.01.

In Figure 8, the average CPU time apparently decreases with the increase of each relaxation factor. It illustrates that these relaxation factors have a significant effect on the CPU time.

Based on the above analysis, we primarily investigate three combinations of them: {"0.001-0.005-0.005",

"0.002-0.005-0.005", "0.02-0.01-0.01"}. The parameters in a combination represent the relaxation factor of benefit, energy, and memory, respectively. The result is shown in Figures 9 and 10.

In Figure 9, the average benefit of "0.001-0.005-0.005" is slightly more than "0.002-0.005-0.005" and the worst is "0.002-0.01-0.01," and the maximum gap between them is less than 0.3%.

In Figure 10, the average CPU time of "0.001-0.005-0.005" is more than "0.002-0.005-0.005" and "0.02-0.01-0.01." The larger the relaxation factors, the more paths removed and the less CPU time.

6.2.2. The Effect of the Lower Bound Cutting Strategy. We first describe the computation of the discount function $f_i(x)$ mentioned in Section 5.1.2. Based on the amount of training data (the same size as the test data), we compute $f_i(x)$ by regression analysis. The process is as follows:

- (1) Select one problem from the training data set.
- (2) Let $P' = \{p'_1, p'_2, \dots, p'_s\}$ (p'_i is the selected subpath in subgraph G_i) be the solution of CPLEX solver (used as the optimal solution) and $P^* = \{p^*_1, p^*_2, \dots, p^*_s\}$ be the solution of GS-BC (used as the estimate solution). The relational data $(x_k, y_k), k \in [1, s]$, is recorded.

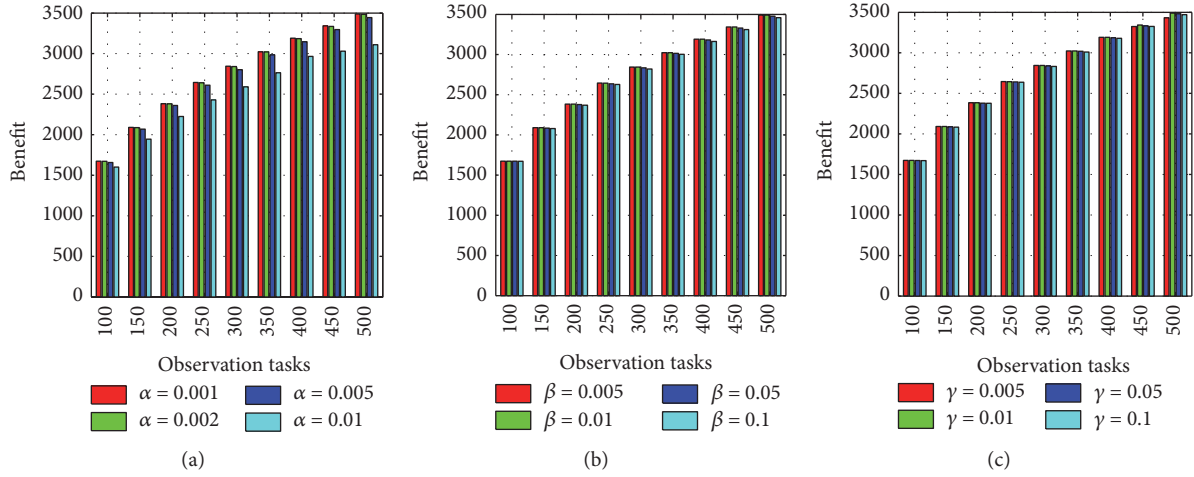


FIGURE 7: The average benefit of LS under single relaxation factor.

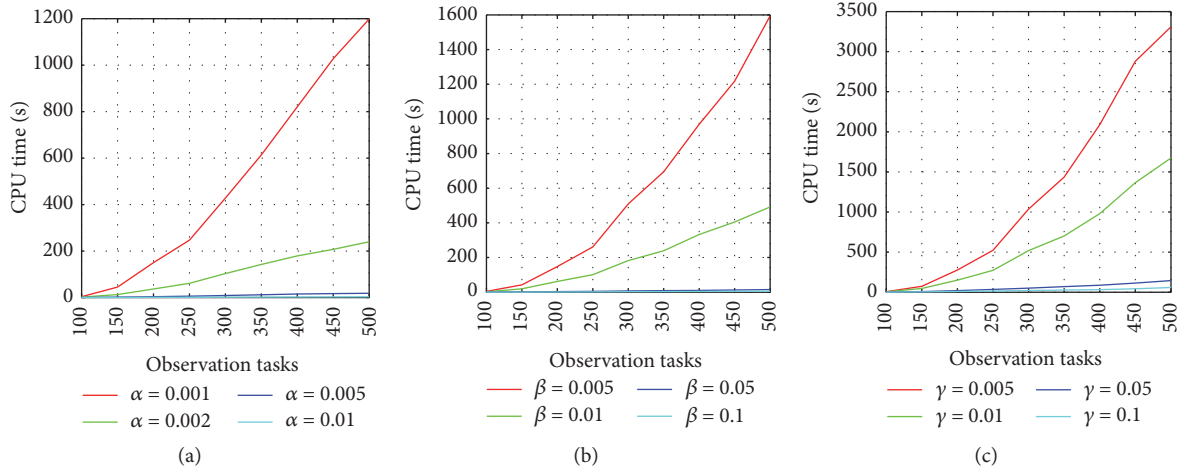


FIGURE 8: The average CPU time of LS under single relaxation factor.

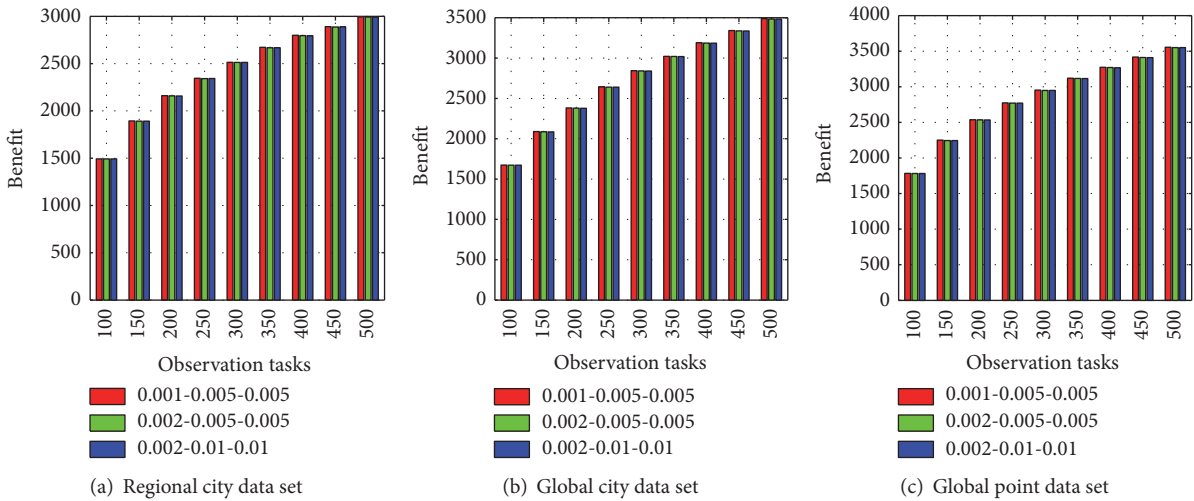


FIGURE 9: The average benefit of LS under the combinations of relaxation factors.

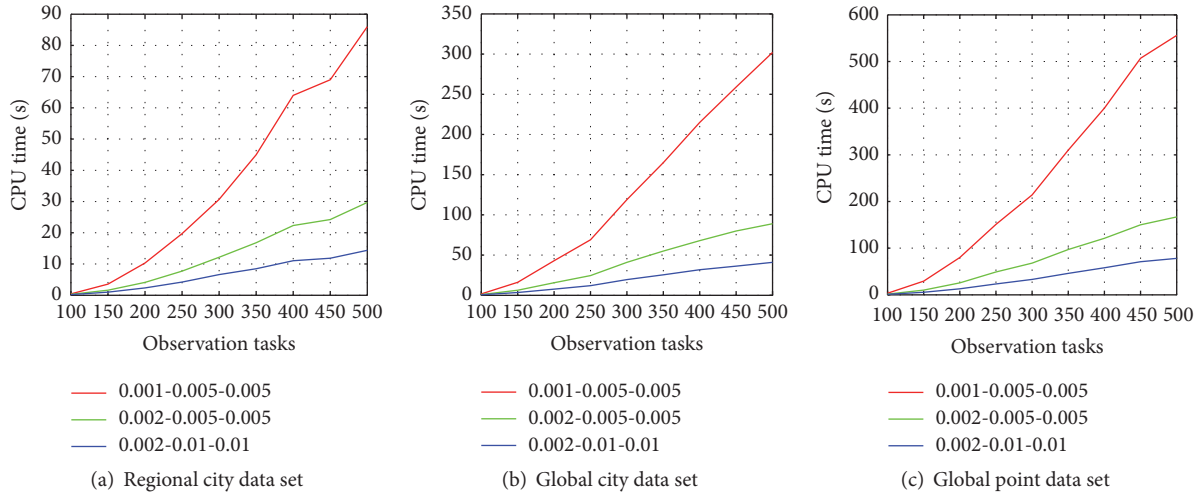
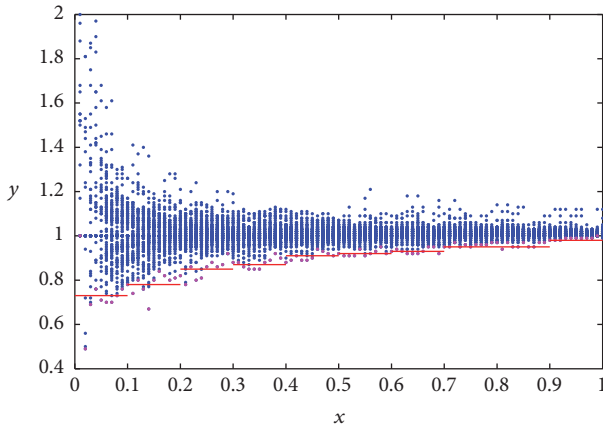


FIGURE 10: The average CPU time of LS under the combinations of relaxation factors.


 FIGURE 11: The distribution of the relational data (x_k, y_k) .

$$(x_k, y_k) = \left(\frac{\sum_{i=1}^k b_i^*}{\sum_{i=1}^s b_i^*}, \frac{\sum_{i=1}^k b_i'}{\sum_{i=1}^s b_i'} \right), \quad (29)$$

where b_i^* and b_i' are the benefit of subpaths p_i^* and p_i' , respectively. The relational data describes the relation between the estimate solution and optimal solution at a subgraph.

- (3) Return to (1) until all problems are solved.
- (4) Compute the regression curve of the lower boundary of the relational data.

Figure 11 shows the distribution of the relational data and the approximate regression curve of the lower boundary, where the blue point is the relational data, the mauve point is the lower boundary, and the red line is $f_1(x)$. In fact, a piecewise function is used as the approximate expression of $f_1(x)$ and confirmed to be effective.

Next, we analyze the effect of the lower bound cutting strategy in LSLBC. The average CPU time is shown in Figure 12. The average benefit is almost the same as Figure 9 that it is not shown here.

Comparing Figures 10 and 12, the CPU time is greatly reduced by using the lower bound cutting strategy. It is because the strategy removes a large number of low-benefit paths during searching.

Through the above analysis, we finally select $\alpha = 0.002$, $\beta = 0.005$, and $\gamma = 0.005$ (e.g., the combination “0.002-0.005-0.005”) as the parameters of LSLBC.

6.3. Comparison Study. Three algorithms including the improved genetic algorithm (IGA) with a stochastic greedy conflict repairing strategy [5], the hierarchical algorithm [7] (called GA-BC), and the CPLEX solver [31] are used as baselines for comparison in this subsection. The stochastic greedy conflict repairing strategy in the IGA is slightly modified to meet the needs of our problem. In the repairing strategy, the observation task with the lowest benefit is removed when the solution violates constraints (6) and (7) until all conflicts are resolved. In the hierarchical algorithm, the observation tasks are firstly planned by the genetic algorithm [3]; then the transmission tasks are scheduled by the branch and cut algorithm [7] based on the selected observation tasks in the previous stage.

The parameters of genetic algorithm in IGA and GA-BC are set as follows: the population size is 50, the crossover probability is 0.8, the mutation probability is 0.008, and the iterative number is 10000. If the solution is not improved in 500 iterations, then the algorithm stops. In CPLEX solver, the gap is set as 0.01, the stop time is set as 5 hours, and other parameters are executed in the default settings. All experiments are repeated 50 times. The following information is recorded; note that these statistics are computed only considering the problems that are solved by all algorithms.

- (i) *ave.b*: the average benefit of an algorithm in a problem set. For example, the average benefit of LSLBC in a problem set is calculated as $(1/10) \sum_{k=1}^{10} b_k(\text{LSLBC})$, where $b_k(\text{LSLBC})$ is the (average) benefit of LSLBC on the k th problem in the problem set.

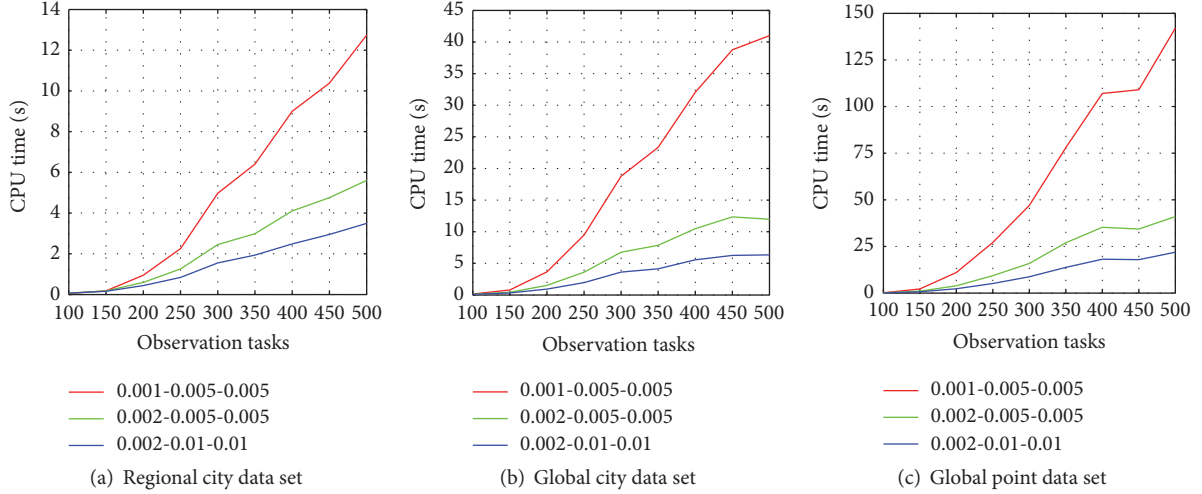


FIGURE 12: The average CPU time of LSLBC under the combinations of relaxation factors.

TABLE 1: The computational result on the regional city data set.

Observation tasks	Problems	CPLEX		LSLBC			IGA			GA-BC		
		ave. <i>b</i>	ave. <i>t</i>	ave. <i>b</i>	ave. <i>t</i>	ave. <i>g</i>	ave. <i>b</i>	ave. <i>t</i>	ave. <i>g</i>	ave. <i>b</i>	ave. <i>t</i>	ave. <i>g</i>
100	10	1501	26.21	1490	0.067	0.70%	1484	3.777	1.09%	1493	3.654	0.56%
150	10	1902	80	1891	0.175	0.62%	1869	7.11	1.73%	1875	7.873	1.42%
200	10	2168	192	2156	0.598	0.55%	2126	10.51	1.91%	2136	11.34	1.46%
250	10	2350	416	2341	1.256	0.37%	2302	12.97	2.02%	2299	15.42	2.17%
300	10	2522	700	2510	2.454	0.46%	2473	16.02	1.95%	2475	17.22	1.87%
350	10	2685	1134	2666	2.987	0.72%	2627	18.66	2.17%	2629	21.10	2.09%
400	10	2810	2205	2794	4.106	0.59%	2744	20.86	2.34%	2748	26.26	2.21%
450	10	2900	3634	2881	4.755	0.67%	2839	23.75	2.11%	2845	28.74	1.91%
500	10	3010	4433	2990	5.603	0.68%	2941	26.04	2.30%	2944	32.46	2.22%

(ii) *ave.t*: the average CPU time of an algorithm in a problem set, measured in seconds. For example, the average CPU time of LSLBC in a problem set is calculated as $(1/10) \sum_{k=1}^{10} t_k(\text{LSLBC})$, where $t_k(\text{LSLBC})$ is the (average) CPU time of LSLBC on the k th problem in the problem set.

(iii) *ave.g*: the average benefit gap between an algorithm and the CPLEX solver. For example, the average benefit error of LSLBC in a problem set is calculated as $(1/10) \sum_{k=1}^{10} ((b_k(\text{CPLEX}) - b_k(\text{LSLBC}))/b_k(\text{CPLEX}))$.

Computational results on three data sets are given in Tables 1, 2, and 3.

In Tables 1, 2, and 3, the benefit of LSLBC is better than IGA and GA-BC, except in a few problems that the number of observation tasks is 100 in the regional city data set and the global point data set. The average CPU times of LSLBC are all less than IGA and GA-BC in the above three data sets. This is because the two preprocessing strategies and the lower bounding cutting strategy in LSLBC remove a large number of redundant paths (the nondominated paths and the low-benefit paths). Though the IGA have a strong global search capability, the local search capability

is weak. Besides, the stochastic greedy repair strategy does not consider the correlation between the observation tasks and the transmission tasks in resource consumption and acquisition. The GA-BC breaks up the optimal solution of the observation tasks during scheduling the transmission tasks.

Tables 1, 2, and 3 show that the average benefit gap of LSLBC is from -0.08% to 0.72% in three data sets. As the number of observation tasks increases, the number of edges in the graph model grows dramatically. The CPLEX solver has to take longer in the preprocessing, which even makes five problems unsolved (in Table 3). In the experiment, the gap of CPLEX solver is 1% which means the solution is not optimal. The LBLSC is an appropriate algorithm that it may be better than CPLEX solver in some problems when the relaxation factors are small, for example, some problems in the global point data set when the number of observation tasks is 200, 250, and 500 (in Table 3). Though the CPLEX solver can find a better solution for most problems, the CPU time is too long that it is difficult to meet the real requirement. The result verifies the ability of LSLBC to solve the SOTTP.

When comparing Tables 1, 2, and 3, we can know that the benefit gap of LSLBC in the global point data set is the least, the next is in the global city data set, and the largest is in the

TABLE 2: The computational result on the global city data set.

Observation tasks	Problems	CPLEX		LSLBC			IGA			GA-BC		
		ave. <i>b</i>	ave. <i>t</i>	ave. <i>b</i>	ave. <i>t</i>	ave. <i>g</i>	ave. <i>b</i>	ave. <i>t</i>	ave. <i>g</i>	ave. <i>b</i>	ave. <i>t</i>	ave. <i>g</i>
100	10	1680	37.92	1669	0.110	0.64%	1661	4.699	1.08%	1668	5.203	0.69%
150	10	2090	132	2080	0.422	0.48%	2041	8.278	2.31%	2076	9.542	0.65%
200	10	2384	371	2377	1.519	0.30%	2314	12.56	2.92%	2361	14.66	0.96%
250	10	2644	722	2636	3.605	0.30%	2570	16.69	2.79%	2619	19.45	0.95%
300	10	2843	1390	2838	6.751	0.16%	2759	21.44	2.97%	2812	24.43	1.09%
350	10	3023	2082	3015	7.846	0.24%	2923	24.98	3.31%	2991	30.34	1.04%
400	10	3193	3956	3183	10.48	0.32%	3088	29.58	3.29%	3154	33.23	1.21%
450	10	3350	7146	3334	12.34	0.49%	3227	33.97	3.69%	3303	40.00	1.43%
500	10	3496	7207	3482	11.98	0.39%	3354	38.54	4.06%	3445	42.26	1.44%

TABLE 3: The computational result on the global point data set.

Observation tasks	Problems	CPLEX		LSLBC			IGA			GA-BC		
		ave. <i>b</i>	ave. <i>t</i>	ave. <i>b</i>	ave. <i>t</i>	ave. <i>g</i>	ave. <i>b</i>	ave. <i>t</i>	ave. <i>g</i>	ave. <i>b</i>	ave. <i>t</i>	ave. <i>g</i>
100	10	1785	28.98	1778	0.124	0.36%	1765	5.779	1.11%	1784	6.559	0.02%
150	10	2246	117	2243	1.039	0.16%	2206	11.87	1.80%	2240	14.95	0.26%
200	10	2532	341	2532	3.893	0	2471	17.26	2.42%	2522	23.96	0.40%
250	10	2768	671	2768	9.349	0.02%	2704	22.91	2.34%	2752	31.80	0.59%
300	10	2949	1441	2944	15.84	0.15%	2871	28.38	2.66%	2932	41.77	0.59%
350	10	3118	3217	3113	26.95	0.16%	3037	34.75	2.59%	3096	48.86	0.67%
400	10	3268	7086	3266	35.25	0.08%	3187	42.05	2.50%	3250	60.00	0.57%
450	10	3415	6873	3406	34.33	0.28%	3318	48.71	2.85%	3385	67.14	0.87%
500	10(5) ¹	3475	11471	3478	41.30	-0.08% ²	3381	56.89	2.70%	3449	74.34	0.74%

¹There are 5 problems unsolved by CPLEX; ²LSLBC is better than CPLEX in some problems.

regional city data set. This reveals that our LSLBC has better performance on decentralized targets.

7. Conclusion

In this paper, we study the single-satellite observation and transmission tasks planning problem under a more accurate resource usage model. Two preprocessing strategies including graph partition and nondominated subpaths selection are used to decompose the problem, and an improved label-setting algorithm with the lower bound cutting strategy is used to solve the problem, which avoids exploring the whole nondominated path set and accelerates the solving procedure. The optimality and feasibility of the proposed method are validated on three data sets, totaling 270 problems. The results demonstrate that our method can find the near-optimal solution in much less time and indicate the ability of our method to solve the SOTTP under different task size.

The future work of our study is mainly focused on two aspects: (1) extending our approach to the scheduling of an agile satellite, which not only selects the tasks to be performed, but also determines the start time of them; (2) extending our approach to the onboard task planning problem for an autonomous satellite. The onboard dynamic situations are considered, such as the arrival of new tasks and cloud disturbances.

Appendix

A. The Computation of the Resource Vector

We define a set of intervals within the period $[t_1, t_2]$; whenever time t is an extreme point of $\pi_{e+}(t) - \pi_{e-}(t) - \phi_{e-}(t)$ or $\phi_d(t)$, a new interval is started (as is shown in Figure 3). This indicates that the energy level and data level are monotonously increasing or decreasing in each interval.

Let $\{t'_0, t'_1, \dots, t'_w\}$ be the set of time points in the period $[t_1, t_2]$, where w is the number of intervals, $t'_0 = t_1$, $t'_w = t_2$. The resource vector of plan p can be calculated by

$$\Delta d_{\max} = \max \left\{ \sum_{k=1}^l \Delta d(t'_{k-1}, t'_k), 1 \leq l \leq w \right\},$$

$$\Delta e_{\max} = \max \left\{ \sum_{k=1}^l \Delta e(t'_{k-1}, t'_k), 1 \leq l \leq w \right\},$$

$$\Delta d_{\min} = \min \left\{ \sum_{k=1}^l \Delta d(t'_{k-1}, t'_k), 1 \leq l \leq w \right\},$$

$$\Delta e_{\min} = \min \left\{ \sum_{k=1}^l \Delta e(t'_{k-1}, t'_k), 1 \leq l \leq w \right\},$$

$$\Delta d_{\text{end}} = \sum_{k=1}^w \Delta d(t'_{k-1}, t'_k),$$

$$\Delta e_{\text{end}} = \sum_{k=1}^w \Delta e(t'_{k-1}, t'_k). \quad (\text{A.1})$$

The energy level and data level at each time point can be calculated by

$$e(t'_l) = \min \{e(t'_{l-1}) + \Delta e(t'_{l-1}, t'_l), E_{\max}\}, \quad (\text{A.2a})$$

$$1 \leq l \leq w,$$

$$d(t'_l) = \max \{d(t'_{l-1}) + \Delta d(t'_{l-1}, t'_l), D_{\min}\}, \quad (\text{A.2b})$$

$$1 \leq l \leq w.$$

B. The Proof of Theorem 2

Proof. In Appendix A, $\forall l \in [1, w]$, it must satisfy the constraint $e(t'_l) \geq E_{\min}$. We can get the following relation from formula (A.2a):

$$\begin{aligned} e(t'_l) &\geq E_{\min}, \quad 1 \leq l \leq w, \\ &\iff e(t'_{l-1}) + \Delta e(t'_{l-1}, t'_l) \geq E_{\min}, \\ &\iff \begin{cases} E_{\max} + \Delta e(t'_{l-1}, t'_l) \geq E_{\min} \\ \vdots \\ E_{\max} + \Delta e(t'_1, t'_l) \geq E_{\min} \\ e(t'_0) + \Delta e(t'_0, t'_l) \geq E_{\min}, \end{cases} \\ &\iff \begin{cases} E_{\max} + \Delta e(t'_0, t'_l) - \Delta e(t'_0, t'_{l-1}) \geq E_{\min} \\ \vdots \\ E_{\max} + \Delta e(t'_0, t'_l) - \Delta e(t'_0, t'_1) \geq E_{\min} \\ e(t'_0) + \Delta e(t'_0, t'_l) \geq E_{\min}, \end{cases} \\ &\iff \begin{cases} \max \{\Delta e(t'_0, t'_a), 1 \leq a \leq l\} - \Delta e(t'_0, t'_l) \leq E_{\max} - E_{\min} \\ e(t'_0) + \Delta e(t'_0, t'_l) \geq E_{\min}. \end{cases} \end{aligned} \quad (\text{B.1})$$

From (A.1) and formula (A.2a), the energy constraints of plan p can be described by the following formulation:

$$\begin{aligned} e(t'_l) &\geq E_{\min}, \quad \forall l \in [1, w], \\ &\iff \begin{cases} \max \{\Delta e(t'_0, t'_a), 1 \leq a \leq l\} - \Delta e(t'_0, t'_l) \leq E_{\max} - E_{\min}, \quad 1 \leq l \leq w \\ e(t'_0) + \min \{\Delta e(t'_0, t'_l), 1 \leq l \leq w\} \geq E_{\min}, \end{cases} \\ &\iff \begin{cases} \max \{\Delta e(t'_0, t'_a), 1 \leq a \leq l\} - \Delta e(t'_0, t'_l) \leq E_{\max} - E_{\min}, \quad 1 \leq l \leq w \\ e(t'_0) + \Delta e_{\min} \geq E_{\min}, \end{cases} \\ &\iff \begin{cases} \max \{\Delta e(t_1, t'), t' \in [t_1, t]\} - \Delta e(t_1, t) \leq E_{\max} - E_{\min}, \quad t \in [t_1, t_2] \\ e(t_1) + \Delta e_{\min} \geq E_{\min}. \end{cases} \end{aligned} \quad (\text{B.2})$$

From (A.1) and formula (A.2b), the memory constraint of plan p is similarly equivalent to the following formulation:

$$\begin{aligned} & - \min \{\Delta d(t_1, t'), t' \in [t_1, t]\} + \Delta d(t_1, t) \\ & \leq D_{\max} - D_{\min}, \quad t \in [t_1, t_2], \end{aligned}$$

$$d(t_1) + \Delta d_{\max} \leq D_{\max}. \quad (\text{B.3})$$

□

C. The Proof of Theorem 3

Proof. According to Appendix A, let $f(l) = e(t'_0) + \sum_{k=1}^l \Delta e(t'_{k-1}, t'_k)$ and $f(l)$ reaches the maximum when $l = l_0$.

If $f(l_0) < E_{\max}$, $\forall l \in [1, w]$, $e(t'_l) = f(l) < E_{\max}$, then

$$\begin{aligned} e(t_2) &= e(t'_w) = e(t'_0) + \sum_{k=1}^w \Delta e(t'_{k-1}, t'_k) \\ &= \min \{E_{\max}, e(t_1) + \Delta e_{\max}\} \\ &\quad - (\Delta e_{\max} - \Delta e_{\text{end}}) \\ &= \min \{E_{\max} - \Delta e_{\max}, e(t_1)\} + \Delta e_{\text{end}}. \end{aligned} \quad (\text{C.1})$$

If $f(l_0) \geq E_{\max}$, then $e(t'_{l_0}) = E_{\max}$ and $e(t'_l) < E_{\max}$, $l_0 < l \leq w$, such that

$$\begin{aligned} e(t_2) &= e(t'_w) = E_{\max} + \sum_{k=l_0+1}^w \Delta e(t'_{k-1}, t'_k) \\ &= E_{\max} - \left(\sum_{k=1}^{l_0} \Delta e(t'_{k-1}, t'_k) - \sum_{k=1}^w \Delta e(t'_{k-1}, t'_k) \right) \\ &= \min \{E_{\max}, e(t'_0) + \Delta e_{\max}\} \\ &\quad - (\Delta e_{\max} - \Delta e_{\text{end}}) \\ &= \min \{E_{\max} - \Delta e_{\max}, e(t_1)\} + \Delta e_{\text{end}}. \end{aligned} \quad (\text{C.2})$$

Similarly, the data level at the end time of subgraph G_i is computed by the following equation:

$$d(t_2) = \max \{D_{\min} - \Delta d_{\min}, d(t_1)\} + \Delta d_{\text{end}}. \quad (\text{C.3})$$

□

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (61101184 and 61174159).

References

- [1] W. Lin, D. Liao, C. Liu, and Y. Lee, "Daily imaging scheduling of an earth observation satellite," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 35, no. 2, pp. 213–223, 2005.
- [2] J. Jang, J. Choi, H.-J. Bae, and I.-C. Choi, "Image collection planning for Korea multi-purpose SATellite-2," *European Journal of Operational Research*, vol. 230, no. 1, pp. 190–199, 2013.
- [3] M. A. A. Mansour and M. M. Dessouky, "A genetic algorithm approach for solving the daily photograph selection problem of the spot5 satellite," *Computers Industrial Engineering*, vol. 58, no. 3, pp. 509–520, 2010.
- [4] S. A. Chien, M. Johnston, J. Frank et al., "A generalized timeline representation, services, and interface for Automating Space Mission Operations," in *Proceedings of the 12th International Conference on Space Operations, SpaceOps 2012*, Sweden, June 2012.
- [5] H. Chen, J. Wu, W. Shi, J. Li, and Z. Zhong, "Coordinate scheduling approach for EDS observation tasks and data transmission jobs," *Journal of Systems Engineering and Electronics*, vol. 27, no. 4, Article ID 7669701, pp. 822–835, 2016.
- [6] F. Khafa, X. Herrero, A. Barolli, L. Barolli, and M. Takizawa, "Evaluation of struggle strategy in genetic algorithms for ground stations scheduling problem," *Journal of Computer and System Sciences*, vol. 79, no. 7, pp. 1086–1100, 2013.
- [7] S. Spangelo, J. Cutler, K. Gilson, and A. Cohn, "Optimization-based scheduling for the single-satellite, multi-ground station communication problem," *Computers Operations Research*, vol. 57, pp. 16–10, 2015.
- [8] V. Gabrel and C. Murat, "Mathematical Programming for Earth Observation Satellite Mission Planning," in *Operations Research in Space and Air*, vol. 79 of *Applied Optimization*, pp. 103–122, Springer US, Boston, MA, 2003.
- [9] G. H. Wu, J. Liu, M. H. Ma, and D. S. Qiu, "A two-phase scheduling method with the consideration of task clustering for earth observing satellites," *Computers Operations Research*, vol. 40, no. 7, pp. 1884–1894, 2013.
- [10] X. W. Wang, Z. Chen, and C. Han, "Scheduling for single agile satellite, redundant targets problem using complex networks theory," *Chaos, Solitons & Fractals*, vol. 83, pp. 125–132, 2016.
- [11] M. Vasquez and J.-K. Hao, "Upper bounds for the SPOT 5 daily photograph scheduling problem," *Journal of Combinatorial Optimization*, vol. 7, no. 1, pp. 87–103, 2003.
- [12] F. Marinelli, S. Nocella, F. Rossi, and S. Smriglio, "A Lagrangian heuristic for satellite range scheduling with resource constraints," *Computers & Operations Research*, vol. 38, no. 11, pp. 1572–1583, 2011.
- [13] L. Barbulescu, J.-P. Watson, L. D. Whitley, and A. E. Howe, "Scheduling space-ground communications for the air force satellite control network," *Journal of Scheduling*, vol. 7, no. 1, pp. 7–34, 2004.
- [14] D. Habet, M. Vasquez, and Y. Vimont, "Bounding the optimum for the problem of scheduling the photographs of an agile earth observing satellite," *Computational optimization and applications*, vol. 47, no. 2, pp. 307–333, 2010.
- [15] M. Vasquez and J.-K. Hao, "A "logic-constrained" knapsack formulation and a tabu algorithm for the daily photograph scheduling of an Earth observation satellite," *Computational optimization and applications*, vol. 20, no. 2, pp. 137–157, 2001.
- [16] R. Xu, H. P. Chen, X. L. Liang, and H. M. Wang, "Priority-based constructive algorithms for scheduling agile earth observation satellites with total priority maximization," *Expert Systems with Applications*, vol. 51, pp. 195–206, 2016.
- [17] 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.
- [18] V. Gabrel and 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*, vol. 139, no. 3, pp. 533–542, 2002.

- [19] D. Karapetyan, S. Mitrovic Minic, K. T. Malladi, and A. P. Punnen, "Satellite downlink scheduling problem: A case study," *OMEGA - The International Journal of Management Science*, vol. 53, pp. 115–123, 2015.
- [20] A. J. Vazquez and R. S. Erwin, "On the tractability of satellite range scheduling," *Optimization Letters*, vol. 9, no. 2, pp. 311–327, 2015.
- [21] Y. Li, R. Wang, Y. Liu, and M. Xu, "Satellite range scheduling with the priority constraint: An improved genetic algorithm using a station ID encoding method," *Chinese Journal of Aeronautics*, vol. 28, no. 3, pp. 789–803, 2015.
- [22] X. S. Jin, J. Li, J. Wang, and N. Jing, "On permutation-based integrated scheduling for earth observing system," *Control Theory & Applications*, vol. 27, no. 4, pp. 407–414, 2010.
- [23] Y. H. Guo, J. Li, and J. Wang, "Key technology research of imaging scheduling system for optical satellite with data transmission," *Journal of National University of Defense Technology*, vol. 30, no. 4, pp. 47–52, 2008.
- [24] J. Li, H. Chen, and N. Jing, "A data transmission scheduling algorithm for rapid-response earth-observing operations," *Chinese Journal of Aeronautics*, vol. 27, no. 2, pp. 349–364, 2014.
- [25] P. Wang, G. Reinelt, P. Gao, and Y. Tan, "A model, a heuristic and a decision support system to solve the scheduling problem of an earth observing satellite constellation," *Computers & Industrial Engineering*, vol. 61, no. 2, pp. 322–335, 2011.
- [26] A. Kennedy, A. Marinan, K. Cahoy et al., "Automated resource-constrained science planning for the mirata mission," in *Proceedings of the 29th Annual AIAA/USU Conference on Small Satellites*, pp. 1–16, 2015.
- [27] G. Beaumet, G. Verfaillie, and M.-C. Charneau, "Feasibility of autonomous decision making on board an agile Earth-observing satellite," *International Journal of Computational Intelligence*, vol. 27, no. 1, pp. 123–139, 2011.
- [28] W.-H. Zhao, J. Zhao, S.-H. Zhao et al., "Resources scheduling for data relay satellite with microwave and optical hybrid links based on improved niche genetic algorithm," *Optik - International Journal for Light and Electron Optics*, vol. 125, no. 13, pp. 3370–3375, 2014.
- [29] C. Park, S. Lee, and S. Park, "A label-setting algorithm for finding a quickest path," *Computers & Operations Research*, vol. 31, no. 14, pp. 2405–2418, 2004.
- [30] H. Chen, J. Li, N. Jing, X. Liu, and Y. Tang, "Scheduling model and algorithms for autonomous electromagnetic detection satellites," *Hangkong Xuebao/Acta Aeronautica et Astronautica Sinica*, vol. 31, no. 5, pp. 1045–1053, 2010.
- [31] IBM ILOG CPLEX optimization studio, <http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/index.html>.



Hindawi

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

