IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received February 6, 2019, accepted February 23, 2019, date of publication March 4, 2019, date of current version May 22, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2902669

A Heuristic Algorithm Based on Temporal Conflict Network for Agile Earth Observing Satellite Scheduling Problem

PING XIE[1,2], HAO WANG[2], YUNING CHEN[2], AND PEI WANG[3]

[1]China Aerospace Academy of Systems Science and Engineering, Beijing, China
[2]College of System Engineering, National University of Defense Technology, Changsha, China
[3]Unit 75831, People's Liberation Army, Guangzhou, China

Corresponding author: Yuning Chen (cynnudt@hotmail.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 71701203 and Grant 61603400.

ABSTRACT Agile Earth Observing Satellite (AEOS) scheduling problem consists of selecting a subset of tasks and developing observation plans for a set of agile satellite resources with the purpose of maximizing the total reward of arranged mission observations. This problem has attracted much attention in recent years since AEOS is a new generation satellite being developed all over the world. Due to its NP-hardness, heuristic methods are widely adopted when solving the AEOS scheduling problem (AEOSSP). In this paper, we propose a temporal conflict network-based heuristic algorithm (TBHA), for AEOSSP. The novelty of TBHA lies in the fact that the heuristics are extracted from a temporal conflict network, which characterizes the overlaps (conflicts) of the visible time windows of the problem. These heuristics are highly effective since they well address the time window conflicts which otherwise pose a significant challenge on the choice of the imaging start time for satellite observations. The extensive simulation experiments with the comparison to a number of heuristic algorithm variants and sophisticated meta-heuristic algorithms are conducted to show that the TBHA algorithm performs very well in terms of both solution quality and computational efficiency.

INDEX TERMS Agile satellite scheduling, complex network, heuristic algorithm, time window conflict.

I. INTRODUCTION

Agile Earth Observing Satellite (AEOS) scheduling problem consists of selecting a subset of tasks and developing observation plans for a set of agile satellite resources with the purpose of maximizing the total reward of arranged mission observations.

Compared to non-agile satellites that have only one degree of freedom, agile satellites are able to move on three axes: roll, pitch and yaw. Due to these new agility capabilities, the azimuth and the starting time of the image acquisition are now free (within given limits), which leads potentially infinite number of ways of acquiring an Earth surface target, giving rise to a better efficiency of the whole satellite system. Fig. 1 shows the comparison between non-agile and agile satellites. In this figure, visible time window (VTW) is a specified time period during which the satellite can take image of the ground target, while observation time window (OTW) defines the actual time interval for the execution of the imaging.

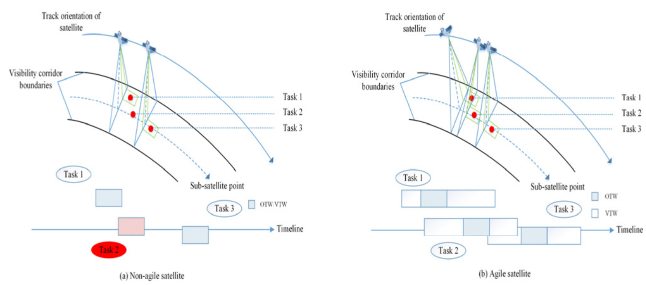

FIGURE 1. Comparisons of non-agile and agile satellites.

AEOS scheduling problem belongs to the NP-hard family, since it generalizes the EOS scheduling problem which was proved to be NP-hard in [1]. The complexity is increased mainly on two aspects. First, the visible time window of the satellite with respect to a given target is extended, which means the starting time of the observation can be any time

The associate editor coordinating the review of this manuscript and approving it for publication was Tao Zhang.

within an interval. Second, the transition between two consecutive observations is time-dependent, which basically means the start time of the latter observation depends on the end time of the former observation.

Due to the NP-hardness of the AEOS scheduling problem, exact methods are only applicable to small-size instances. As a result, heuristics and meta-heuristics are often considered in the literature. Lemaître [2] provided four methods including a greedy algorithm, a dynamic programming procedure, a constraint programming method, and a local search method, to solve a simplified version of the scheduling problem for AEOSs; Cordeau and Laporte [3] inherited the problem model of [2] and developed a tabu search method to solve the scheduling problem of a given orbit; Dilkina and Havens [4] investigated different local optimization algorithms (including hill-climbing, simulated annealing and squeaky wheel optimization) coupled with constraint propagation for handling image acquisition time windows; Liao and Yang [5] developed an imaging order scheduler for the FORMOSAT-2 satellite and pointed out that the AEOS scheduling problem is characterized by sequence-dependent setups. They proposed a rolling scheduling framework to deal with weather uncertainty. They used Lagrangian relaxation and linear search techniques to solve the problem; Li *et al.* [6] described a combined genetic algorithm for selecting and scheduling the AEOS scheduling problem; Habet *et al.* [7] formulated the AEOS problem as a constrained optimization problem whose main objective function is to maximize the profit, and the secondary objective is to minimize the sum of transition durations. They proposed a tabu search algorithm to solve it; Yuning *et al.* [8], Guo *et al.* [9], Yan *et al.* [10] and Zhang *et al.* [11] proposed three different ant colony algorithms for the AEOS scheduling problem; Tangpattanakul *et al.* [12] presented a biased random-key genetic algorithm to solve a multi-objective optimization problem associated with the management of AEOSs; Xu *et al.* [13] proposed priority-based constructive algorithms with total priority maximization. Liu *et al.* [14] proposed an adaptive large neighborhood search meta-heuristic for AEOS scheduling problem.

Compared to common approaches that solve the AEOS scheduling problem directly based on its mathematical formulation, it is a new point of view to study the problem from the perspective of complex network. To this end, we can take each possible observation opportunity as a network node, and the relationship or connection of them are described by network edges. Useful information can then be extracted from the network which helps us to analyze the problem and develop effective heuristic algorithms. The idea of this work is inspired by a number of papers applying complex network theory to the field of production scheduling [15]–[18]. However, in the area of AEOS scheduling (or more generally EOS scheduling), there are very few such works. A small number of somewhat related works are summarized below. Zufferey *et al.* [19] adapted the best ingredients of the graph coloring techniques to the NP-hard satellite range scheduling problem, called MuRRSP, then a tabu search and an adaptive memory algorithm were designed to tackle it. Sarkheyli *et al.* [20] followed this line in investigating the scheduling system of low earth orbit (LEO) satellite operations. Wang *et al.* [21] firstly introduced the theory of complex networks and found similarities between AEOS redundant targets scheduling problem and the node centrality ranking problem, then proposed a fast approximate scheduling algorithm (FASA) to obtain effective scheduling results. We note in the above papers that, the edge of the network models is represented by the order of tasks (two adjacent tasks are connected by an edge). This modeling method is no longer effective in the setting of AEOS scheduling problem since the prolonged visible time window (VTW) is no longer equal to the observation time window (OTW). Though Wang *et al.* [21] tries improving this modeling method by granulating the original VTW into several candidate OTWs, but it should be noticed that the solution accuracy highly depends on the granularity interval. Moreover, they did not consider the transition time and the observation quality in their paper. These are in fact two very essential factors and cannot be ignored in AEOS scheduling problem. Our work attempts to address the above issues. We recall that the most prominent characteristic of an agile satellite is its long visible time window with respect to a ground target. This usually causes many overlaps between visible time windows of different tasks as Fig. 2, which poses significant challenge on the choice of starting time for these tasks especially when the number of tasks is large. Therefore, it is more reasonable to use the overlaps between visible time windows to reflect the edge connection relationship. This results in a so-called temporal conflict network (TCN) which is first proposed in this work. In addition, our AEOS scheduling problem is a more realistic model that takes transition time (between two consecutive observations) and the observation quality into account.

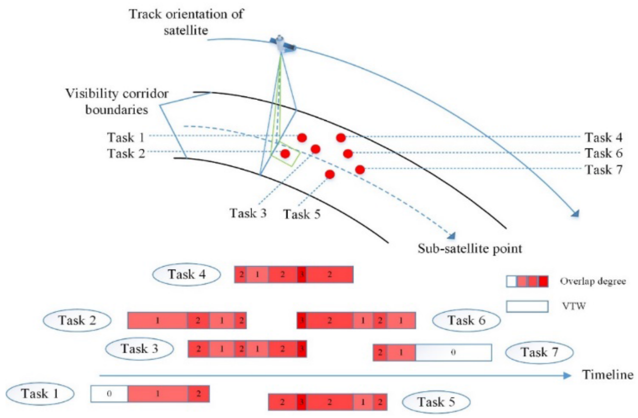

FIGURE 2. Diagrammatic sketch of VTW overlaps.

In this paper, we attempt to extract a set of highly effective heuristic rules from the TCN of AEOS scheduling problem. These heuristic rules are then integrated into a construction algorithm, leading to a TCN based heuristic algorithm (TBHA), which is able to produce high quality solutions for the AEOS scheduling problem. Experimental results on a set of representative instances show that the proposed TBHA can achieve better results than construction algorithms using basic rules, and even the most recent state-of-the-art meta-heuristic algorithms ALNS in [14].

The remainder of this paper is organized as follows. Section II provides a formal definition of the AEOS scheduling problem. Section III presents the temporal conflict network (TCN) and its propagation effect. Section IV describes a TCN based heuristic algorithm for solving the AEOS scheduling problem. Computational experiments are presented in Section V, followed by conclusions in Section VI.

II. PROBLEM DESCRIPTION

The elements of AEOS scheduling problem can be described as follows: there are a set of satellites and a set of observing tasks; each task has a priority; each task corresponds to a set of available visible time windows.

AEOS scheduling problem can be described as under various constraints, selecting part of observing tasks from a series of ground targets which need to be observed, and then arranging observing tasks and observing time according to a number of continuous orbits for agile satellites, and finally maximizing the total observation reward.

Before formulating the AEOS scheduling problem, we made a number of reasonable assumptions:

1) The satellite has only one sensor;
2) All tasks are point targets;
3) Energy and storage are both unlimited;

The main variables and parameters used in this paper are defined in Section A.

A. VARIABLES

- $T = \{t_1, t_2, \text{ł}; dots, t_n\}$: the set of tasks, where n is the number of tasks to be scheduled.
- p_i : the priority level of the task t_i, which measures the benefit earned by fulfilling it.
- θ_i : the specified observation angle for task t_i.
- q_i : the observation quality of task t_i, which is affected by the specified observation angle θ_i of the task.
- g_i : the prospective profit of observing task t_i, which is determined by the priority level and the observation quality.
- d_i : the duration time needed for task t_i to get a total observation.
- $TW_i = \{tw_{i1}, tw_{i2}, \ldots, tw_{im}\}$: the set of visible time window for task t_i, where tw_{ij} represents the jth visible time window for task t_i.
- ws_{ij}, we_{ij}: the start time and end time for visible time window tw_{ij}.
- o_i : the observation time window for task t_i.
- os_i, oe_i: the start time and end time of observation time window for task t_i.
- tr_{ij} : the transition time needed for the satellite to transit the posture from the end observing angle θ_i to of the former task t_i to the start observing angle θ_j of the next task t_j.

B. MATHEMATICAL FORMULATION

In this section, we formally describe the AEOS problem. First, we define a binary decision variable x_{ij}:

$$x_{ij} = \begin{cases} 1, & o_i \subseteq tw_{ij} \\ 0, & otherwise \end{cases} \tag{1}$$

Equation (1) shows if visible time window tw_{ij} is selected to arrange the observation time window for task t_i, then x_{ij}= 1, else x_{ij}= 0. And the symbol $\subseteq$ is redefined here as the affiliation of time window. We write $o_i \subseteq tw_{ij}$ if and only if $ws_{ij} \leq os_i \leq oe_i \leq we_{ij}$.

To better describe sequence-dependent structure of the AEOS scheduling problem, we define a binary auxiliary variable $F_{ij} \in (0, 1)$, $t_i, t_j \in T$, where F_{ij}= 1 represents that task t_i and task t_j are observed continuously.

The optimization goal of the AEOS scheduling problem is to maximize the sum of the observation profits, which can be expressed as follows:

$$Max \sum_{i=1}^{|T|} \sum_{j=1}^{|TW_i|} x_{ij} g_i \tag{2}$$

Subject to:

$$\sum_{j=1}^{|TW_i|} x_{ij} \leq 1, \quad \forall t_i \in T \tag{3}$$

$$oe_i = os_i + d_i, \quad \forall t_i \in T \tag{4}$$

$$ws_{ij} \leq os_i \leq oe_i \leq we_{ij}, \quad x_{ij}= 1 \tag{5}$$

$$oe_i + tr_{ij} \leq os_j, \quad \forall t_i, t_j \in T \wedge F_{ij} = 1 \tag{6}$$

$$g_i = p_i \cdot \sqrt{q_i}, \quad \forall t_i \in T \tag{7}$$

Equation (3) requires that each task can be observed at most once. In other words, for each task we can choose only one visible time window to fulfill its observation request.

Equation (4) indicates that each observation activity must be ensured d_i time for the completeness of the observation.

Equation (5) requires the observation time window to be within the visible time window.

Equation (6) requires the time between two adjacent observations must be sufficient for the maneuver of the satellite. The time required for maneuver transition is calculated by:

$$tr_{ij} = \begin{cases} \Delta\theta_{ij}/v_1 + \lambda_1, & \Delta\theta_{ij} \leq \Delta\theta_{t_1} \\ \Delta\theta_{ij}/v_2 + \lambda_2, & \Delta\theta_{t_1} < \Delta\theta_{ij} \leq \Delta\theta_{t_2} \\ \Delta\theta_{ij}/v_3 + \lambda_3, & \Delta\theta_{ij} > \Delta\theta_{t_2} \end{cases} \tag{8}$$

As common, we adopt a piecewise linear function to represent the transition time. In equation (8), $\Delta\theta_{ij}$ represents the total angle change between two adjacent observation o_i and o_j, v represents the transition speed of satellite, λ represents the transition stabilization time which is related to $\Delta\theta_{ij}$, and $\Delta\theta_t$ represents the threshold of $\Delta\theta_{ij}$.

There is no well recognized revenue model for the AEOS scheduling problem considering image quality. We define a rough revenue model as equation (7), showing the actual profit of the task is determined by the priority of the task and the quality of the observation. We define the range of the priority as $p_i \in [1, 10]$, and the observation quality q_i is calculated as follows:

$$q_i = 10 - 9\frac{\left|\frac{os_i+oe_i}{2} - \frac{ws_{ij}+we_{ij}}{2}\right|}{\frac{we_{ij}-ws_{ij}}{2} - \frac{oe_i-os_i}{2}} \tag{9}$$

As $q_i \in [1, 10]$, the dimension of the observation quality is the same with p_i. According to the characteristics of agile satellites, the best image quality is achieved with the nadir pointing of the satellite, i.e, the OTW is exactly in the middle of the VTW. As can be seen from equation (9), when the OTW is located in the middle of the VTW, the observation quality is 10; while the observation quality is 1 when it is on both sides.

III. TEMPORAL CONFLICT NETWORK FOR HEURISTIC EXTRACTION

Following most of the existing works, we propose to solve the AEOS scheduling problem using heuristic algorithms, where heuristic information is extracted from a temporal conflict network (TCN) that fully characterizes the "conflict" relationship (overlaps) of all candidate visible time windows (VTWs).

In the AEOS scheduling problem considered in this paper (see the problem description of Section II), an AEOS typically has multiple VTWs (that lie in different tracks) with respect to a ground target, which usually have many overlaps (conflicts) with VTWs of nearby ground targets, due to their extended length provided by the satellite agility. This makes the selection of VTWs for a selected task become significantly more difficult. We seek to extract useful local and global information from TCN to guide the VTW selection, allowing highly informed decisions.

A. BASIC TCN CONSTRUCTION

A temporal conflict network G can be represented by a two-tuple, that is $G = (V, E)$, where V represents a set of nodes and E represents a set of edges. These network elements are described as follows:

- Node set V: A node corresponds to a visible time window;
- Edge set E: Two edges exist between two nodes if their corresponding time windows are overlapped. Each edge is associated with a specific weight.

Fig. 3 shows the framework of a temporal conflict network with three tasks. As we can see, corresponding to the running orbit, the satellite can have multiple VTWs for each observation task. These VTWs are nodes in the temporal conflict network, and their overlap relationships are represented by edges. It should be noted that there is only one undirected edge between each pair of VTWs in this figure, while in

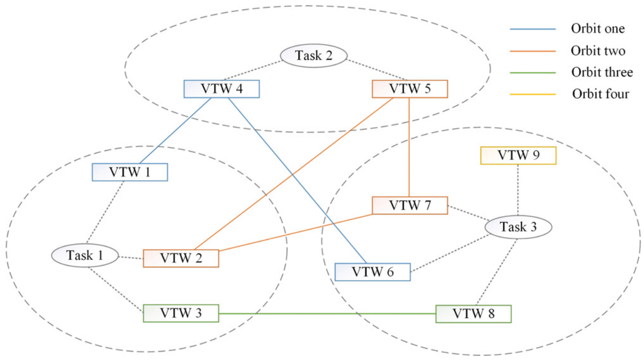

FIGURE 3. Framework of temporal conflict network (TCN).

the real TCN (as we can see below), the undirected edge is divided into two directed edges with specific weights.

In this paper, a directed edge from the source node i to the target node j represents the conflict influence of node i on the visible window of node j. Since the actual observation time window (OTW) is just a part of the selected VTW, there is only one hidden mapping between the VTW and the actual OTW, so we proposed a rough method to represent the conflict influence. The weight of the edge is represented by the product of the overlap proportion of the target node's total visible time window with the target node's task priority:

$$e_{ij} = p_j \cdot \frac{\Delta\left|we_{i\cup j} - ws_{i\cup j}\right|}{we_j - ws_j} \tag{10}$$

where $\Delta\left|we_{i\cup j} - ws_{i\cup j}\right|$ denotes the overlap interval between the VTW i and j.

Based on the TCN, the weighted out-degree of a task reflects its impact on the neighboring tasks and the weighted in-degree indicates the influence it suffered from its neighboring tasks.

$$\langle W_{out}\rangle_i = \sum_j e_{ij} \tag{11}$$

$$\langle W_{in}\rangle_i = \sum_j e_{ji} \tag{12}$$

where $\langle W_{out}\rangle_i$ represents the weighted out-degree of VTW i, and $\langle W_{in}\rangle_i$ represents the weighted in-degree of it.

To maximize the total reward, the nodes having high weighted in-degree should be considered first because they are likely to have high priority and high conflicts (overlaps) with other nodes. On the other hand, the nodes having low weighted out-degree should be considered first when it comes to the VTW selection for a ground target. Based on the out-degree and in-degree of nodes, we define the time window sensitivity (TWS) and time window influence (TWI) as follows:

$$TWS_i = \frac{\langle W_{out}\rangle_i}{\sum_{i=1}\langle W_{out}\rangle_i} \tag{13}$$

$$TWI_i = \frac{\langle W_{in} \rangle_i}{\sum\limits_{i=1} \langle W_{in} \rangle_i} \tag{14}$$

TWS and *TWI* are further used to define an indicator called target priority (*TP*):

$$TP_i = \frac{\sum\limits_{j=1}^{|TW_i|} TWS_j}{count\,(|TW_i|)} \tag{15}$$

where $count\,(|TW_i|)$ denotes the number of VTWs associated with target i. *TP* is used to sort tasks in our proposed heuristic algorithm (see Section IV), which determines the order in which tasks are considered.

In the heuristic algorithm proposed in Section IV, the *TP* value is used to determine a task processing order and the nodes out-degree value is used to decide a VTW consideration order. VTWs with lower out-degrees have smaller impact on others, leaving more opportunities for other tasks to be included in the final solution.

B. TCN STRUCTURE IMPROVEMENT

The basic TCN models the conflicts among VTWs of different tasks. However, it does not consider the mutually exclusive relationship of the different VTWs of the same task. The improved TCN includes additional edges to describe such relationship. Compared to the overlaps of VTWs that belong to different tasks, the overlaps of the mutually exclusive VTWs are more pronounced, whose corresponding edges should be given a higher weight:

$$e_{ij} = \alpha p_j \tag{16}$$

where α denotes the weight coefficient, and VTW i and VTW j belong to the same task.

Fig. 4 and Fig. 5 (based on the 50-target scenario in Section V) show the structural difference between the basic TCN and the improved TCN.

As we can see from Fig. 4 and Fig. 5, the basic TCN has several isolated parts corresponding to different satellite tracks while the improved TCN is a fully connected graph. And this difference will affect the propagation effect in Section C.

C. PROPAGATION EFFECT OF TCN

While we arrange the observation time window according to the timeline, it is obvious that the influence of time window conflict is a cumulative process. A task may be not only influenced by its neighboring tasks, but also other tasks that have a path in connection with it. In other words, the impact of a node may propagate to another node via some immediate nodes in between. Thus, the time window sensitivity (*TWS*) and time window influence (*TWI*) should be calculated by considering not only the neighboring nodes (out-degree and in-degree), but also nodes along path. In this section, we put forward a proper way to consider the propagation effect in our temporal conflict network.

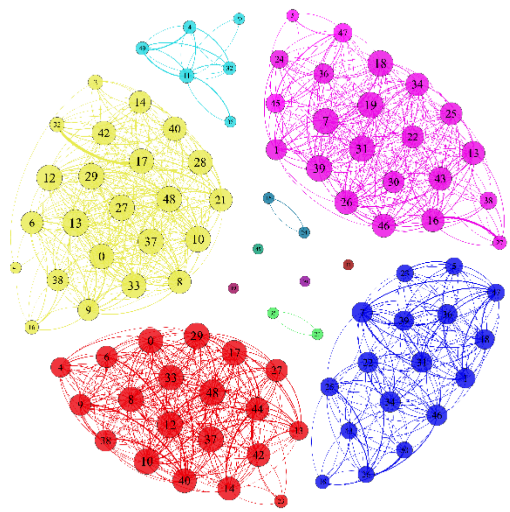

FIGURE 4. The basic temporal conflict network.

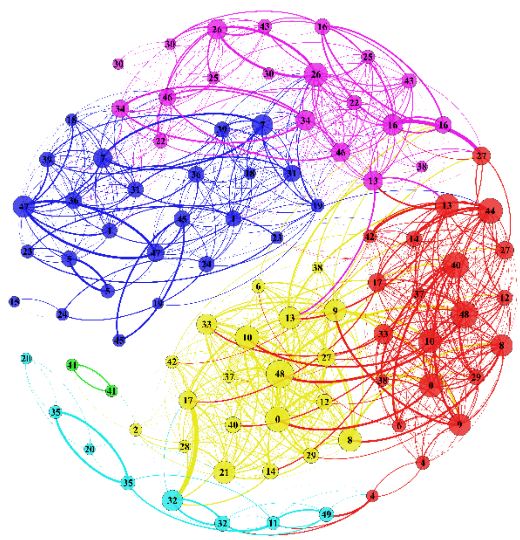

FIGURE 5. The improved temporal conflict network.

As we mentioned in Section I, the AEOS scheduling problem is similar to the node centrality ranking problem [21]. In the field of complex network, the propagation effect between nodes is usually an unignorably factor while performing node centrality ranking. To consider this effect in our TCN, we propose to use a well-known node centrality ranking algorithm called PageRank (PR) [22].

PR algorithm is firstly used in web page ranking for Google, which mainly includes two rules: First, when a web page is linked by more web pages, its ranking will be higher; second, when a web page is linked by a high-ranking web page, its importance should be enhanced. Based on these two rules, the ranking of a web page is equal to the weighted ranking of all pages linked to the web page. In other words, the standard for ranking is changed from the basic linking number to the sum of weighted linking number. This is how PR algorithm considers the propagation effect of the linked web pages.

We draw lessons from the PR algorithm to handle the propagation effect of the TCN and then re-calculate the *TWS* and *TWI*. To better use the PR model, we use *TW* to collectively denote both *TWS* and *TWI* and define the initial importance of the node as follows:

$$p_i^0 = TW_i \tag{17}$$

The iterative formula for PR value is:

$$p^{n+1} = d \cdot E_c \cdot p^n + (1 - d) \cdot p^0 \tag{18}$$

where p^n represents the PR value matrix after the nth iteration, E_c represents the edge weight matrix after column normalized (for *TWS*) or row normalized (for *TWI*), $p^0 = \{p_1^0, p_2^0, \ldots, p_n^0\}$ represents the node initial importance matrix, d represents damping coefficient, and $d \cdot E \cdot A$ represents that the node in the random model will own a certain share of the PR value by the weight matrix to each outside the chain.

The framework of the PR algorithm is shown below:

Step 1: Calculate the initial importance matrix of nodes and the column weight matrix after normalization;

Step 2: Calculate the PR value after iteration;

Step 3: Normalize the PR value;

Step 4: Terminate the algorithm if the maximum number of iterations has been reached, turn to Step 2 otherwise.

The PR algorithm terminates when a maximum number of iterations are reached.

IV. A TCN BASED HEURISTIC ALGORITHM FOR AEOS SCHEDULING PROBLEM

In this section, we present a TCN based heuristic algorithm (TBHA) for solving AEOS scheduling problem. TBHA is essentially a multi-start construction algorithm, which relies on heuristics extracted from the TCN for the guidance of solution construction. The pseudo-code of the TBHA algorithm is shown in Algorithm 1.

TBHA basically iterates for four runs, each run using a different network. We recall in Section III that the basic TCN can be improved by two strategies: ST1) Establishing connections between VTWs of the same task; ST2) Leveraging propagation effect. This leads to four types of networks, which are named N1 (basic TCN), N2 (basic TCN+ST1), N3 (basic TCN+ST2), and N4 (basic TCN +ST1+ST2). Based on our preliminary experiments, none of the four networks has a dominant performance over others in any case. The proposed TBHA thus integrates all four networks to obtain high quality solutions and to enhance its generality.

The inputs of TBHA are: a network set N, a task set T, and a visible time window set W. Based on equation (15), we use function RankT(T, st) to rank the task set T and keep the new sorted task list in T'. According to the task order in T', we try to insert the tasks into the solution S one by one. And for each task t, we use function RankW(W_t) to rank all its VTWs according to equation (14). The resulting VTWs are stored in W_t'. Then we try to arrange the starting time in each VTW according to W_t', and find the best choice w_{best} for the specific task with function $f(w)$ (see equation (7)). The w_{bes} is also included into the solution S. The best of the four resulting solutions under four different networks is the output of the TBHA.

Algorithm 1 Pseudo-code of the TBHA algorithm

```
Input:
Strategy set: N = {N1, N2, N3, N4}; Task set: T =
{t1, t2, ..., tn}; VTW set: W = {w1, w2, ..., wn};
//Initialization:
Sbest ← Ø;
fbest ← 0;
//Main Loop:
for tcn in N
  S ← Ø;
  T' ← RankT (T, tcn); /* Rank the task list basing on TP
value from TCN */
  for t in T'/*W_t is the VTW list corresponding to task t*/
     W_t' ← RankW (W_t); /* Rank the VTW list basing on
TWI value from TCN */
     fbest ← 0;
     for w in W_t'
       if (the observation time window can be arranged in w)
        if (fbest < f(w)) /* Comparison of VTW profit */
        fbest← f(w);
           w_best ← w;
       end
      end
     end
     if(fbest != 0 )
        S ← S ⋃ w_best; /*w_best is the arranged VTW with
ProfitTbest */
     end
  end
  if( fbest < f(S)) /* Comparison of scheduling plan profit */
     Sbest ← S;
  end
end
ReturnSbest
```

V. COMPUTATIONAL EXPERIMENTS

In order to assess the effectiveness of the proposed algorithm, we carried out extensive numerical experiments on a set of well-designed scenarios, whose results will be presented and discussed in this section.

A. DESIGN OF TESTED SCENARIOS

The algorithm was coded in C#, and the experiments were conducted using Intel (R) Core (TM) i7-6700HQ CPU 2.60 GHz under Windows 10 with 8 GB RAM.

TABLE 1. Orbit parameters of the satellite.

Satellite	a	e	i	ω	Ω	m
AS-01	7141701.7	0.000627	98.5964	95.5069	342.307	125.2658

The configurations of the tested scenarios are illustrated below. Due to the differences of the associated mission, design and adopted technologies, the satellites used in different countries differ significantly in terms of capabilities, constraints and management. It is therefore unreasonable to use a common benchmark to test a scheduling algorithm designed for a specific satellite. No comparative data and no competing heuristic exist for this problem and optimal solutions are not available.

In the paper, requests are generated by a random uniform distribution on China's territory. Each request is associated with five factors: priority, geographical position, duration of observation, image due time, and specified image angle for some tasks. The position of each task is defined by latitude, longitude and altitude. Targets are distributed in a predefined region, within mainland China, and are defined by the area corresponding to 3° N–53° N and 74° E–133° E. Under this mode, 15 scenarios were generated, and the number of targets contained in these scenarios varies from 50 to 400, with an increment step of 25.

We used the same satellite AS-01 as in [14], whose largest pitch degree is 45°, roll degree being 45° and yaw degree being 90°. The time horizon of the scheduling is 24 h, from 2017/07/20 00:00:00 to 2017/07/20 24:00:00, where the satellite is allowed to fly for multiple tracks. The satellite's location in space is characterized by six orbital parameters which are the length of semi-major axis (a), eccentricity (e), inclination (i), argument of perigee (ω), right ascension of the ascending node (Ω), and mean anomaly (m). The initial orbital parameters for AS-01 are shown in Table 1.

Recall that there are a number of parameters associated with the proposed temporal conflict network (TCN). Their settings are as follows:

- The transition speed of satellite (°/s): $v_1 = v_2 = v_3 = 2$;
- The transition stabilization time (s): $\lambda_1 = 0$; $\lambda_2 = 5$; $\lambda_3 = 10$;
- The threshold of angle transition (°):$\Delta\theta_1 = 10$; $\Delta\theta_2 = 20$;
- The weight coefficient of the mutually exclusive VTWs:$\alpha = 1.7$;
- The max iteration of PageRank value: 2;
- The damping coefficient of PageRank value:$d = 0.85$;

B. EXPERIMENTAL RESULTS OF TBHA

Table 2 shows the results obtained for the scenarios obtained with different annealing parameters. The columns VTWs and T_v show the number of visible time windows and tasks of each scenario. The column T_s shows the number of tasks arranged by the TBHA algorithm. R_s is the ratio of arranged tasks over all tasks. Similarly, R_p is the ratio of the sum profit of arranged tasks over all tasks. The column P_v is the total profit of all tasks and P_s is the total profit of arranged tasks. We also report the CPU time in the column "Seconds".

TABLE 2. Computational Results of the TBHA algorithm.

Scenario	VTWs	T_v	T_s	R_s	P_v	P_s	R_p	Seconds
50	93	49	48	0.979592	279	261.16	0.936067	0.043
75	144	74	63	0.851351	378	335.61	0.886926	0.08195
100	176	98	78	0.795918	576	480.54	0.829852	0.139
125	237	124	95	0.766129	656	523.32	0.797743	0.184
150	276	148	97	0.655405	849	617.62	0.727473	0.313
175	330	174	110	0.632184	1007	708.45	0.703529	0.3785
200	360	194	108	0.556701	1097	678.42	0.618432	0.4545
225	411	223	126	0.565022	1337	784.73	0.586934	0.5365
250	453	247	127	0.51417	1460	832.23	0.570019	0.6455
275	494	271	132	0.487085	1578	805.58	0.510506	0.733
300	551	296	141	0.476351	1690	858.33	0.507885	0.898
325	606	319	145	0.454545	1864	953.57	0.511573	1.0805
350	642	350	139	0.397143	1792	867.23	0.483947	1.2055
375	704	372	149	0.400538	2108	967.82	0.459118	1.4345
400	730	394	150	0.380711	2135	955.97	0.447762	1.61

TABLE 3. Comparative results of TBHA with other rule-based heuristic algorithms.

Scenario	R	LPT	SPT	G	E	G-LPT	G-SPT	TBHA
50	248.00	242.62	231.90	256.58	249.31	257.77	258.32	**261.16**
75	264.31	295.08	287.05	326.48	301.87	326.60	334.55	**335.61**
100	436.88	389.87	408.91	445.68	378.36	457.67	466.82	**480.54**
125	413.48	377.09	409.66	501.34	396.65	503.79	515.90	**523.32**
150	495.13	504.47	486.41	569.80	483.20	580.49	594.80	**617.62**
175	546.36	538.80	552.52	650.16	531.35	661.61	651.11	**708.45**
200	560.14	465.93	538.84	663.51	499.47	679.69	**684.85**	678.42
225	624.83	584.67	612.89	736.65	614.09	733.99	753.83	**784.73**
250	612.30	607.62	636.28	774.50	621.12	797.24	790.57	**832.23**
275	580.54	579.99	609.13	747.43	566.07	761.18	779.49	**805.58**
300	628.07	637.09	729.37	852.80	651.22	848.39	**872.06**	858.33
325	679.22	641.35	723.49	889.59	703.72	884.77	895.55	**953.57**
350	596.79	582.90	644.43	850.12	603.66	842.12	853.38	**867.23**
375	682.65	702.97	685.67	955.84	643.50	**974.78**	963.91	967.82
400	691.65	644.53	663.60	908.79	649.43	942.75	935.15	**955.97**
Total	8060.35	7794.98	8220.14	10129.26	7893.02	10252.83	10350.30	10627.70

TABLE 4. Different ranking indicators for the seven rule-based heuristic algorithm (HA).

Algorithm	R	LPT	SPT	G	E	G-LPT	G-SPT
TP_i	——	d_i	$1/d_i$	p_i	p_i/d_i	$p_i - d_i$	$p_i - 1/d_i$

From these results, we can see that in general the number of acquired tasks grows as the increase of the number of user requests. The growth slows down in the last five scenarios (from 300 to 400) where we find that the number of scheduled tasks changes slightly. This is because the available imaging time is highly constrained by the trajectory of the satellite. It is the bottleneck of the satellites service capacity that is most constrained. Another phenomenon we observe is that the profit ratio is always larger than the completion ratio except for the 50 tasks scenario, which can be explained by the fact that the algorithm first schedules tasks with higher priorities.

C. COMPARISON WITH RULE-BASED HEURISTIC ALGORITHM VARIANTS

Table 3 compares the results of the proposed TBHA algorithm with 7 widely used rule-based heuristic algorithm, including random (R) algorithm, longest processing time first (LPT) algorithm, shortest processing time first (SPT) algorithm, greedy (G) algorithm, efficiency first (E) algorithm, greedy with longest processing time first (G-LPT) algorithm, and greedy with shortest processing time first (G-SPT) algorithm.

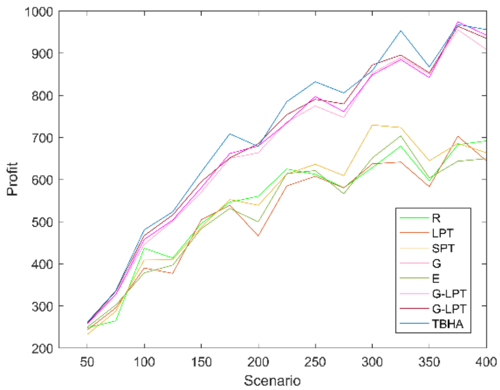

FIGURE 6. Comparison with rule-based heuristic algorithms.

The main difference of those algorithm lies in the strategy used to rank the task set (Line 10 of Algorithm 1). In Table 4, we show the ranking indicator, that is the target priority (TP) and symbol "-" connects two indicators, e.g., for G-LPT, we first sort tasks by priority (p_i) and then sort tasks with the same priority with their duration time (d_i).

As we can see in Fig. 6, it is clear that the results of R, SPT, LPT and E algorithm are much worse than G, G-LPT, and G-SPT algorithm. This means heuristic algorithms based on priority first rule can achieve good results for AEOS scheduling problem. Also, it is worth noting that the E algorithm doesn't obtain good result as opposed to G-LPT and G-SPT. This is because the dimension of the task priority and processing time is not consistent, in which case it is not appropriate to use the ratio of the priority and processing time to evaluate the algorithm efficiency, and that is why we don't consider the task processing time as a heuristic factor in the TBHA algorithm.

For these results, we can see that TBHA heuristic algorithm obtains 12 best results out of 15 scenarios, and the remaining 3 best results are shared by G-SPT and G-LPT. Compared to the commonly used heuristic algorithms, TBHA appears to have a clear advantage. THBA achieves a nearly 4.92 % of improvement over basic greedy algorithm in terms of total profit (10627.70 v.s.10129.26).

D. COMPARISON WITH TWO SOPHISTICATED METAHEURISTIC ALGORITHMS

In order to further verify the performance of the algorithm, we compare the results of TBHA with two metaheuristic algorithms, including the basic ACO algorithm [10] and a more brilliant ALNS algorithm [16]. Each algorithm was run 10 times, and we calculated the maximum, the minimum, and the average of the results. The detail results of each scenario are show in Table 5.

To provide a clear picture of the comparative results of TBHA with ACO and ALNS, we plotted their average results in Fig. 7. We can see that when the task number of scenarios is small, the results of the metaheuristic algorithms are superior to TBHA algorithm. However, when the problem size is large (see those scenarios with over 300 tasks), TBHA shows a dominating performance. This is because, according to our experimental observation, their initial solutions are of poor quality and moreover they have no effective diversification mechanism to jump out of local traps. From this experiment, we conclude that TBHA is highly scalable, showing an excellent performance for large-sized problems.

TABLE 5. Comparative results of TBHA with two metaheuristic algorithms.

Scenario	TBHA	ACO			ALNS		
		best	worst	average	best	worst	average
50	261.16	262.14	260.96	261.74	265.64	259.54	**262.72**
75	335.61	347.44	341.95	344.03	358.52	353.58	**355.87**
100	480.54	490.86	473.93	483.38	521.10	497.97	**511.32**
125	523.32	541.35	517.82	525.93	584.18	533.35	**558.64**
150	617.62	640.95	608.95	623.17	725.57	672.66	**700.14**
175	**708.45**	707.69	683.92	695.12	747.47	670.89	700.47
200	678.42	693.66	672.09	683.23	729.48	609.31	**683.97**
225	**784.73**	778.10	735.72	761.74	797.81	750.87	775.88
250	832.23	842.44	825.18	**833.61**	846.76	785.68	823.94
275	**805.58**	808.40	756.11	784.59	796.97	648.13	739.11
300	858.33	877.74	841.02	**860.99**	881.83	836.94	858.33
325	**953.57**	948.80	896.15	921.25	791.54	719.10	752.58
350	**867.23**	879.47	837.76	857.58	809.86	680.58	740.07
375	**967.82**	960.82	935.19	952.23	808.52	707.75	750.86
400	**955.97**	945.11	890.59	909.78	850.55	729.05	780.77

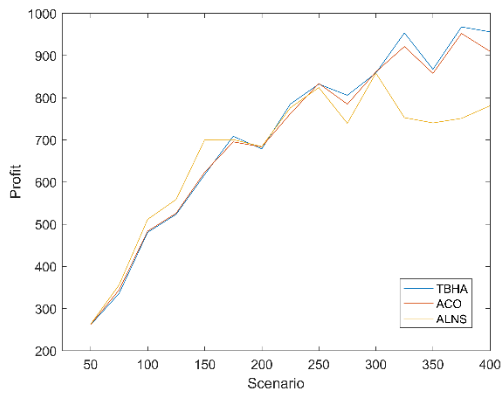

FIGURE 7. Comparison with metaheuristic algorithms.

TABLE 6. Results of the improved ACO with TBHA.

Scenario	ACO			ACO- TBHA			Gap (%)
	best	worst	average	best	worst	average	
50	262.14	260.96	261.74	262.45	262.18	262.30	0.2148
75	347.44	341.95	344.03	347.30	344.10	346.15	0.6155
100	490.86	473.93	483.38	486.00	483.03	484.40	0.2120
125	541.35	517.82	525.93	532.35	523.32	531.26	1.0142
150	640.95	608.95	623.17	635.38	623.29	628.00	0.7741
175	707.69	683.92	695.12	723.07	711.08	716.39	3.0595
200	693.66	672.09	683.23	706.55	698.99	702.51	2.8227
225	778.10	735.72	761.74	798.42	789.81	794.25	4.2669
250	842.44	825.18	833.61	848.59	836.78	841.14	0.9044
275	808.40	756.11	784.59	821.95	814.46	818.45	4.3161
300	877.74	841.02	860.99	903.71	886.79	895.75	4.0379
325	948.80	896.15	921.25	961.11	957.65	959.48	4.1489
350	879.47	837.76	857.58	905.04	875.22	890.25	3.8102
375	960.82	935.19	952.23	986.43	969.33	978.19	2.7264
400	945.11	890.59	909.78	985.09	966.81	976.04	7.2833

E. IMPROVE ACO WITH TBHA

Since TBHA is a construction algorithm in nature, its outcome can be used as a starting point (initial solution) of metaheuristic algorithm. It is interesting to know whether TBHA can help to improve the performance of existing metaheuristic algorithms, we tested the ACO algorithm [10] with its initial solution replaced by the one provided by

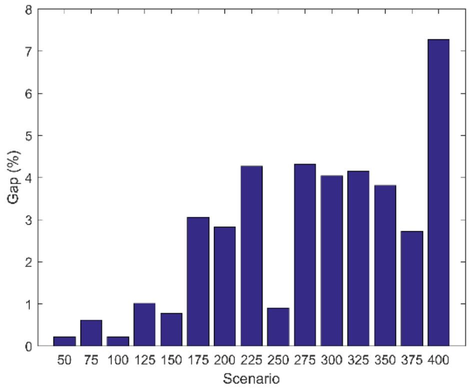

FIGURE 8. Improvement in the ACO.

TABLE 7. Comparative results of TBHA variants with four different strategies.

Scenario	TBHA	N1	N2	N3	N4
50	261.16	**261.16**	**261.16**	260.24	258.67
75	335.61	**335.61**	333.98	335.26	332.96
100	480.54	**480.54**	477.89	478.00	455.43
125	523.32	507.72	508.01	521.37	**523.32**
150	617.62	610.58	611.22	**617.62**	603.43
175	708.45	701.65	**708.45**	693.88	684.46
200	678.42	670.17	671.38	676.33	**678.42**
225	784.73	767.11	**784.73**	774.33	771.92
250	832.23	830.72	**832.23**	830.61	825.13
275	805.58	804.93	**805.58**	**805.58**	804.73
300	858.33	821.58	828.19	828.30	**858.33**
325	953.57	925.33	925.95	928.34	**953.57**
350	867.23	825.19	835.07	854.96	**867.23**
375	967.82	921.97	940.83	937.94	**967.82**
400	955.97	921.07	914.71	916.31	**955.97**
Total	10627.70	10385.33	10439.38	10459.06	10541.40

TBHA, leading to a new ACO named ACO-TBHA. The detailed comparative results of ACO and ACO-TBHA are shown in Table 6 and the improvement gaps are plotted in Fig. 8. It can be seen that ACO-TBHA is able to improve over ACO. From Fig. 8, we can see that the improvement is relatively small (around 1%) when the problem size is small (with less than 150 tasks), and it increases to about 4% when the problem size grows larger. In particular, for the largest 400-task scenario, the improvement jumps up to 7.28%.

F. COMPARISON WITH FOUR SIMPLIFIED TBHAS

The TBHA algorithm is essentially a combination of four different solutions strategies in the framework of the network model. Table 7 compares the results of different strategies. As we defined in Section IV, N1 represents the basic TCN, N2 represents the basic TCN with ST1, N3 represents the basic TCN with ST2, and N4 represents the basic TCN with ST1 and ST2. As can be seen from the results, the algorithm performance of N4 is most excellent, especially after the task number reaches an enough large scale. And from the total profit of those 15 scenarios, we can see that considering the mutually exclusive relations of meta-tasks is better than without it and considering the propagation of time windows overlap effect is better than no consideration, which proves the effectiveness of the improved TCN.

TABLE 8. Comparative results of computational time.

Algorithm	HA	TBHA	ACO	ALNS
Total time (s)	0.8659	9.73745	19018.2	14839.54

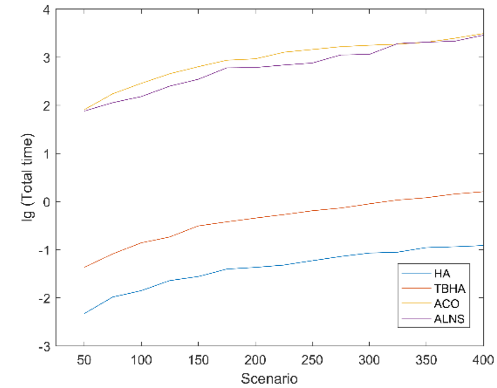

FIGURE 9. Comparison of computational time.

G. COMPARISON OF COMPUTATIONAL TIME

Finally, Table 8 summarizes the total computational time of TBHA algorithm with ACO and ALNS algorithms with all scenarios and Fig. 9 shows the growth trend of computational time across 15 scenarios in the logarithmic coordinate system. As we can see, though TBHA is 9 to 13 times slower than the most basic rule-based heuristic algorithm, but this efficiency dominance is even larger when compared to complicated ACO and ALNS algorithms.

VI. CONCLUSION AND FUTURE WORK

In this paper, we studied agile earth observation satellite scheduling problem and developed a TBHA heuristic algorithm to solve it. The main idea of TBHA is to extract heuristics from a temporal conflict network that features the overlaps of visible time windows, where tasks are taken as network nodes, and the overlaps of their time windows are described by network edges. The weights of edges are based on the priority of tasks and the proportion of the intersection with respect to the overlapped visible time windows. Since a ground target may have more than one visible time window, and only one of them can be selected, special weighted edges are added between nodes in the network. A task may be not only influenced by its neighboring tasks, but also other tasks that have a path in connection with it. Thus, the weights of edges should be calculated by considering not only the two neighboring nodes, but also nodes along path. We use PageRank algorithm to handle this propagation effect and then calculate an improved weighted out-degree and in-degree, which are used as heuristic information to sort tasks in our proposed TBHA.

In simulation experiments, a number of heuristic algorithm variants and two sophisticated meta-heuristic algorithms (ACO and ALNS) are compared to demonstrate the effectiveness of the proposed TBHA algorithm. The results show that THBA easily dominates the heuristic algorithm variants and competes very well with ACO and ALNS. In particular, TBHA shows a clear advantage over ACO and ALNS in large scale scenarios.

Future work should involve additional constraints, such as power supply and memory constrain to the problem model. In addition, data download should also be taken into account. As to the solution method, we would like to investigate dynamic temporal conflict network which may change over time after tasks have been added into the partial solution.

REFERENCES

[1] W. J. Wolfe and S. E. Sorensen, "Three scheduling algorithms applied to the earth observing systems domain," *Manage. Sci.*, vol. 46, no. 1, pp. 148–166, 2000.

[2] M. Lemaître, G. Verfaillie, F. Jouhaud, J.-M. Lachiver, and N. Bataille, "Selecting and scheduling observations of agile satellites," *Aerosp. Sci. Technol.*, vol. 6, no. 5, pp. 367–381, 2002.

[3] J.-F. Cordeau and G. Laporte, "Maximizing the value of an Earth observation satellite orbit," *J. Oper. Res. Soc.*, vol. 56, no. 8, pp. 962–968, 2005.

[4] B. Dilkina and B. Havens. (2005). *Agile Satellite Scheduling Via Permutation Search With Constraint Propagation*. Accessed: May 5, 2014. [Online]. Available: http://www.cs.sfu.ca/CourseCentral/827/havens/Ppapers/topic,12

[5] D. Y. Liao and Y. T. Yang, "Imaging order scheduling of an earth observation satellite," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 37, no. 5, pp. 794–802, Sep. 2007.

[6] Y. Li, M. Xu, and R. Wang, "Scheduling observations of agile satellites with combined genetic algorithm," in *Proc. 3rd Int. Conf. Natural Comput. (ICNC)*, vol. 3, Aug. 2007, pp. 29–33.

[7] D. Habet, M. Vasquez, and Y. Vimont, "Bounding the optimum for the problem of scheduling the photographs of an Agile Earth Observing Satellite," *Comput. Optim. Appl.*, vol. 47, no. 2, pp. 307–333, Oct. 2010.

[8] C. Yuning, X. Lining, and C. Yingwu, "Scheduling of agile satellites based on ant colony algorithm," *Sci. Technol. Eng.*, vol. 11, no. 3, pp. 484–489, 2011.

[9] H. Guo, D.-S. Qiu, G.-H. Wu, and H.-I. Wang, "Tasks scheduling method for an agile imaging satellite based on improved ant colony algorithm," *Syst. Eng.-Theory Pract.*, vol. 32, no. 11, pp. 2533–2539, 2012.

[10] Z. Z. Yan, Y. W. Chen, and L. N. Xing, "Agile satellite scheduling based on improved ant colony algorithm," *Syst. Eng.-Theory Pract.*, vol. 3, p. 28, 2014.

[11] Z. Zhang, N. Zhang, and Z. Feng, "Multi-satellite control resource scheduling based on ant colony optimization," *Expert Syst. Appl.*, vol. 41, no. 6, pp. 2816–2823, 2014.

[12] P. Tangpattanakul, N. Jozefowiez, and P. Lopez, "A multi-objective local search heuristic for scheduling Earth observations taken by an agile satellite," *Eur. J. Oper. Res.*, vol. 245, no. 2, pp. 542–554, 2015.

[13] R. Xu, H. Chen, X. Liang, and H. Wang, "Priority-based constructive algorithms for scheduling agile earth observation satellites with total priority maximization," *Expert Syst. Appl.*, vol. 51, pp. 195–206, Jun. 2016.

[14] X. Liu, G. Laporte, Y. Chen, and R. He, "An adaptive large neighborhood search metaheuristic for agile satellite scheduling with time-dependent transition time," *Comput. Oper. Res.*, vol. 86, pp. 41–53, Oct. 2017.

[15] D. A. Batallas and A. A. Yassine, "Information leaders in product development organizational networks: Social network analysis of the design structure matrix," *IEEE Trans. Eng. Manag.*, vol. 53, no. 4, pp. 570–582, Nov. 2006.

[16] A. Sedeño-Noda, D. A. L. de Pablo, and C. González-Martín, "A network flow-based method to solve performance cost and makespan open-shop scheduling problems with time-windows," *Eur. J. Oper. Res.*, vol. 196, no. 1, pp. 140–154, 2009.

[17] Q. Xuan and T.-J. Wu, "Network model and heuristic scheduling rule designing method for complex open shop problems," *J. Zhejiang Univ. (Eng. Sci.)*, vol. 45, no. 6, pp. 961–968, 2011.

[18] Y. Zhan, Y. G. Zhong, and H. T. Zhu, "Preemptive open-shop scheduling: Network flow based algorithm," *Adv. Mater. Res.*, vol. 215, pp. 111–114, Mar. 2011.

[19] N. Zufferey, P. Amstutz, and P. Giaccari, "Graph colouring approaches for a satellite range scheduling problem," *J. Scheduling*, vol. 11, no. 4, pp. 263–277, 2008.

[20] A. Sarkheyli, B. G. Vaghei, R. A. Moghadam, and A. Bagheri, "Scheduling earth observation activities in LEO satellites using graph coloring problem," in *Proc. 5th Int. Symp. Telecommun.*, Dec. 2010, pp. 928–931.

[21] 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, Feb. 2016.

[22] L. Page, S. Brin, R. Motwani, and T. Winograd, "The PageRank citation ranking: Bringing order to the Web," Stanford InfoLab, Tech. Rep., 1999.

PING XIE is currently pursuing the Ph.D. degree in management science and engineering with the National University of Defense Technology, Changsha, China.

He is also a Researcher with the China Aerospace Academy of Systems Science and Engineering. His current research interests include space planning, satellite engineering demonstration, satellite measurement, and control integration.

HAO WANG received the B.S. degree in management from Liaoning Normal University, Dalian, China, in 2017. She is currently pursuing the master's degree with the National University of Defense Technology, Changsha, China.

Her current research interests include satellite mission planning, combinatorial optimization and reinforcement learning, and data analysis.

YUNING CHEN received the B.Eng. and M.Eng. degrees from the National University of Defense Technology (NUDT), Changsha, China, in 2010 and 2012, respectively, and the Ph.D. degree in computer science from the LERIA Laboratory, University of Angers, France, in 2016. He is currently an Assistant Professor with NUDT. His research interests include developing efficient algorithms for solving photograph satellite scheduling problem, navigation satellite communication optimization problem, and other classical combinatorial optimization problems.

PEI WANG was born in 1982. He received the B.S. degree in system engineering, and the M.S. and Ph.D. degrees in management science and engineering from the National University of Defense Technology, in 2004, 2006, and 2011, respectively.

He was a Visiting Student with the Discrete Optimization Research Group, University of Heidelberg, Germany, from 2008 to 2010. He is currently a Lieutenant Colonel with the Unit 75831, PLA. His research interests include operation mission planning and intelligent optimization algorithms.