Ensemble of Metaheuristic and Exact Algorithm Based on the Divide-and-Conquer Framework for Multisatellite Observation Scheduling

GUOHUA WU, Member, IEEE
QIZHANG LUO
Central South University, Changsha, China, and also National University of Singapore, Singapore

XIAO DU
Central South University, China

YINGGUO CHEN
National University of Defense Technology, Changsha, China

PONNUTHURAI NAGARATNAM SUGANTHAN, Fellow, IEEE
Nanyang Technological University, Singapore, and also KINDI Center for Computing Research, College of Engineering, Qatar University, Qatar

XINWEI WANG
TU Delft, CN, Delft, the Netherlands

Manuscript received 7 September 2021; revised 22 November 2021 and 15 February 2022; released for publication 16 March 2022. Date of publication 22 March 2022; date of current version 11 October 2022.

DOI. No. 10.1109/TAES.2022.3160993

Refereeing of this contribution was handled by P. Singla.

This work was supported in part by the National Natural Science Foundation of China under Grant 61603404 and Grant 71801218 and in part by the Natural Science Fund for Distinguished Young Scholars of Hunan Province under Grant 2019JJ20026.

Authors' addresses: Guohua Wu and Xiao Du are with the School of Traffic and Transportation Engineering, Central South University, Changsha 410075, China, E-mail: (guohuawu@csu.edu.cn, 1272068435@qq.com); Qizhang Luo is with the School of Traffic and Transportation Engineering, Central South University, Changsha 410075, China, and also with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore 119260, E-mail: (qz_luo@csu.edu.cn); Yingguo Chen is with the College of Systems Engineering, National University of Defense Technology, Changsha 410073, China, E-mail: (ygchen@nudt.edu.cn); Ponnuthurai Nagaratnam Suganthan is with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798 and also with KINDI Center for Computing Research, College of Engineering, Qatar University, Qatar, E-mail: (epnsugan@ntu.edu.sg); Xinwei Wang is with the School of Traffic and Transportation Engineering, Central South University, Changsha 410075, China, and also with the Department of Transport and Planning, TU Delft, 2628 CN Delft, the Netherlands. *(Corresponding author: Xinwei Wang.)*

Satellite observation scheduling plays a significant role in improving the efficiency of Earth observation systems. To solve the large-scale multisatellite observation scheduling problem, this article proposes an ensemble of metaheuristic and exact algorithms based on a divide-and-conquer framework (EHE-DCF), including a task allocation phase and a task scheduling phase. In the task allocation phase, each task is allocated to a proper orbit based on a metaheuristic incorporated with a probabilistic selection and a tabu mechanism derived from ant colony optimization and tabu search, respectively. In the task scheduling phase, we construct a task scheduling model for every single orbit and solve the model by using an exact method (i.e., branch and bound, B&B). The task allocation and task scheduling phases are performed iteratively to obtain a promising solution. To validate the performance of the EHE-DCF, we compare it with B&B, three divide-and-conquer-based metaheuristics, and a state-of-the-art metaheuristic. Experimental results show that the EHE-DCF can obtain higher scheduling profits and complete more tasks compared with existing algorithms. The EHE-DCF is especially efficient for large-scale satellite observation scheduling problems.

I. INTRODUCTION

Earth observation satellites (EOSs) are widely used in sensing Earth's surface and surrounding atmosphere. The extremely useful imaging capabilities of EOSs have played an important role in resource exploration, disaster surveillance, urban planning, and environmental monitoring [1], [2]. In recent years, although the number of EOSs is increasing continuously and has reached 906 by January 1st 2021, the satellites are still insufficient for serving numerous Earth observation requests [3]. Therefore, the EOS scheduling problem that aims to accomplish as many observation requests as possible forms an essential component in the EOS systems.

An illustration of the EOS imaging activity is shown in Fig. 1. An EOS flies around the Earth along with its fixed track, and its sensor could generate an observation strip with a certain width and length when passing over a ground target. To observe multiple ground targets, the EOS needs to conduct certain operations for the transfer between two consecutive observation tasks, such as attitude slewing and stabilization. Besides, the EOS can only perform imaging operations within a limited time window [4], [5] when it can pass a ground target and catch sight of the target. During the observation process, each ground target has different visible time windows for different EOSs. Meanwhile, an EOS can observe the same ground target on different orbits. Here, a single orbit is defined as a time interval that the satellite circles the Earth once [6], [7]. The scheduling horizon (24 h in our study) is split into multiple orbits (i.e., intervals), and each orbit includes candidate tasks with observation time windows for certain ground targets. The observation time windows are obtained in advance using satellite-to-target visibility calculations [8]. Thus, there could be multiple visible time windows between an EOS and a ground target. Although many impressive studies have been carried out to address EOS scheduling problems [6], [8]–[11], the increasing number of orbiting satellites and user demands has posed new challenges on multisatellite scheduling problems with large-scale tasks in practical applications. In

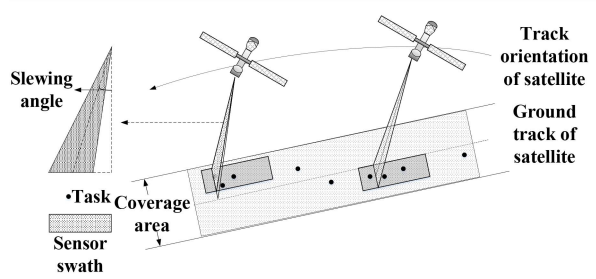

Fig. 1. Satellite imaging activity.

this article, we aim to tackle a multisatellite, multiorbit, and large-scale Earth observation scheduling problem. The difficulties for solving this kind of problem can be viewed in two aspects. First, the number of candidate EOSs and visible time windows for serving a task, as well as the number of tasks could increase the problem complexity exponentially. Second, multiple satellites indicate that the constraints would be more complex to guarantee the collaboration of satellites, which increases the difficulties for solving the problem compared with single satellite scheduling problems. In regard to these difficulties, the exact algorithms developed in the existing literature are no longer suitable for large-scale scheduling problems, as the computational time of exact algorithms is unacceptable [12], [13]. On the other hand, although heuristic and meta-heuristics have been widely used to solve large-scale scheduling problems [14]–[17], the scheduling results of these methods may not have performance guarantees. Therefore, it could be natural and of great significance to develop efficient satellite scheduling methods combing the exact and metaheuristics, which can bring together the advantages of these two types of optimization algorithms to achieve a higher scheduling performance, to satisfy more user demands and improve the efficiency of the Earth observation systems.

Motivated by this, in this study, we develop a scheduling framework based on the well-known divide-and-conquer principle [18], which decomposes a large-scale EOS scheduling problem into multiple subproblems to reduce the complexity of solving the problem. We treat the orbits of satellites as the resources providing imaging services and propose a novel scheduling method under the divide-and-conquer framework (DCF). This method comprises two phases: task allocation among multiple orbits and task scheduling on a single orbit. In the task allocation phase, we develop a metaheuristic allocation method based on the idea of pheromone in ant colony optimization (ACO) and the tabu mechanism in tabu search (TS). After the task allocation phase, multiple subproblems that schedule tasks on each orbit are generated. Afterward, in the task scheduling phase, we construct an integer programming model for each single orbit scheduling problem and utilize a branch-and-bound (B&B) method to solve this model exactly. The task allocation and single orbit scheduling in the two phases are executed iteratively and interactively until a promising solution is obtained.

The overall approach can be viewed as an ensemble of metaheuristic and exact algorithms based on the DCF. It provides a new paradigm that cooperatively uses metaheuristics and exact mathematical programming approaches to solve complex and large-scale combinatorial optimization problems. When confronting a complex and large-scale optimization problem, metaheuristics may not be effective to find a high-quality solution, while exact methods generally cannot obtain an optimal solution with affordable computation time. The proposed approach can partition the original problem into multiple subproblems by using a metaheuristic, and then, solves each simpler subproblem via exact and mature mathematical programming approaches. The obtained solution would be of more high quality, while the optimization process would be more efficient.

The main contributions of this article are summarized as follows:

1) We propose an ensemble of metaheuristic and exact algorithm based on the DCF (EHE-DCF) to address a multisatellite, multiorbit, and large-scale Earth observation scheduling problem. The proposed framework is a new paradigm that decomposes a large-scale scheduling problem into subproblems and combines the advantages of metaheuristic and exact algorithm.
2) In the EHE-DCF, we treat the orbits of satellites as independent resources that could provide imaging services, and divide the scheduling process into two phases, i.e., task allocation among multiple orbits in the first phase and task scheduling on every single orbit in the second phase. These two phases are iteratively and interactively performed to further improve the quality of the solution.
3) We design a metaheuristic based on the pheromone mechanism used in ACO and the tabu mechanism used in TS to realize the effective task allocation in the first phase of the EHE-DCF. In addition, we employ a mathematical model and a corresponding B&B method to address the task scheduling problem on every single orbit in the second phase.
4) Extensive experiments on EOS scheduling instances with multisatellite, multiorbit, and large-scale tasks are conducted to validate the performance of the EHE-DCF. Specifically, the EHE-DCF is compared with five existing approaches: an exact method without the DCF (denote as pure B&B), three DCF-based metaheuristics, i.e., greedy algorithm based on the DCF (GR-DCF), simulated annealing algorithm based on greedy neighborhood structure and DCF (SANS1-DCF), and simulated annealing algorithm based on random neighborhood structure and DCF (SANS2-DCF), as well as a state-of-the-art metaheuristic (ASA) [19]. The experimental results demonstrate the superiority of the EHE-DCF.

The rest of this article is organized as follows. Section II surveys the related works. Section III provides the scheduling framework based on the divide-and-conquer principle. In Section IV, we present a mathematical model for the single orbit task scheduling. We introduce the EHE-DCF in Section V. The simulation experiments and the results are detailed in Section VI. Finally, Section VII concludes this article.

II. RELATED WORKS

At present, considerable achievements have been made in the domain of EOS scheduling problems. The models established by scholars include mathematical programming models [3], [20]–[23], constraint satisfaction problem models [24]–[26], knapsack problems [27]–[29], and graph-based problems [11], [30], [31]. The algorithms for EOS scheduling problems can be roughly classified into exact, heuristic, and metaheuristic approaches.

In general, exact algorithms are feasible in tackling small-scale EOS scheduling problems [32]. For instance, Gabrel and Vanderpooten [7] formulated the EOS scheduling problem as the selection of a multiple criteria path in a graph without a circuit. This problem was solved by the generation of efficient paths and the selection of a satisfactory path using multiple criteria interactive procedure. Hu *et al.* [32] conducted a study on the application of exact algorithms to EOS constellations and proposed a branch-and-price algorithm to solve the EOS constellation imaging and downloading integrated scheduling problem. Peng *et al.* [6] investigated the agile satellite scheduling problem with time-dependent profits and solved the problem based on an adaptive-directional dynamic programming algorithm with decremental state-space relaxation.

Exact algorithms can get optimal scheduling results; but for NP-hard optimization problems, the required computation time of exact algorithms usually increases exponentially with the increase of problem scale. Thus, heuristics and metaheuristics are carried out to solve the EOS scheduling problems. For example, Wu *et al.* [19] developed a formal model for EOS scheduling problems, and presented an adaptive simulated annealing SA-based scheduling algorithm integrated with a dynamic task clustering strategy . Huang *et al.* [33] presented a multiobjective chance constrained programming model for electronic reconnaissance satellites scheduling problem, and proposed a Monte Carlo simulation-based multiobjective evolutionary algorithm. Many scholars have applied the ACO algorithm to solve EOS scheduling problems [34]–[37]. For example, Gao *et al.* [34] constructed an acyclic directed graph model for the EOS scheduling problem and presented a novel hybrid ACO method. Zhang *et al.* [36], [37] presented a complex independent set model for multisatellite control resource scheduling problem, and proposed an ACO-based algorithm, in which the pheromone trail is updated by two stages to avoid local optima. Wang *et al.* [38] established an integer programming model for the EOS scheduling problem, and proposed a hybrid ACO algorithm, where the pheromone was used to indicate how to choose the request to schedule . These existing works inspired us to design a metaheuristic based on the pheromone mechanism of ACO to realize the task allocation in the first phase.

In recent years, a new trend for solving the EOS scheduling problem is to decompose the large-scale scheduling problem into several small-scale scheduling problems that can be solved separately. For instance, Xu *et al.* [39] transferred the very large area observation problems into a set covering problem with constraints and solved the problems based on a three-phase algorithm. Liu *et al.* [40] decomposed the scheduling problem into two subproblems: task assignment and task merging. Our study is distinguished from these studies in two aspects. First, we decompose the scheduling problem into subproblems based on a DCF, which can solve the problem iteratively and interactively. Second, we propose an ensemble of metaheuristic and exact algorithms, which combines the advantages of these two kinds of algorithms, thereby improving the efficiency of the optimization process significantly.

Furthermore, in the aforementioned studies, scholars usually formulated satellites as resources and assumed that each task has at most one observation time window on each resource. However, a satellite could have multiple orbits to provide multiple observation time windows by passing over a ground target multiple times in the scheduling horizon. Hence, the observation time windows for a ground target will not be unique, increasing the difficulties in solving the EOS scheduling problem. As in some existing studies [6], [41], [42], we formulate the satellites orbits as different resources, such that each resource involves at most one observation window for each ground target, making the EOS scheduling problem easier to model. Moreover, we address larger-scale problems compared with the aforementioned studies.

III. DIVIDE-AND-CONQUER-BASED SCHEDULING FRAMEWORK

As Fig. 1 shows, an EOS could generate an observation strip of a certain width and length when passing over a ground target. The width and length of the strip are determined by the altitude of the satellite, as well as the view field, the slewing angle, and the observation duration of the sensor [5]. In order to facilitate modeling, we assume that all ground targets are point targets and a ground target is termed a task. Besides, the orbits of the satellites are termed as resources that could provide imaging services. To schedule the satellite resources efficiently, we propose a novel scheduling framework based on the DCF. The framework comprises two iterative phases: task allocation phase among multiple orbits and task scheduling phase on every single orbit, whose workflow is shown in Fig. 2.

In the task allocation phase, we develop a metaheuristic to allocate tasks to orbits. We first calculate the probability of each task to be allocated to each orbit. The calculation of the allocation probability between a task and an orbit is inspired by the idea of the pheromone mechanism of

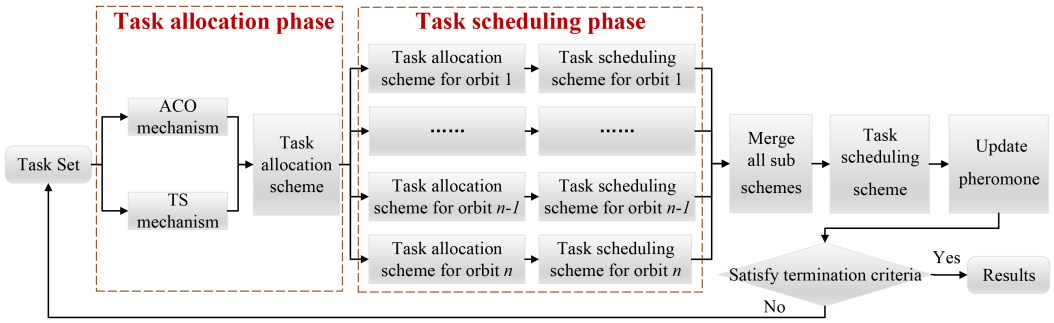

Fig. 2. Scheduling framework based on the divide-and-conquer principle.

ACO [37]. Once a task is allocated to an orbit, the intensity of pheromone between the task and the orbit will be increased. Besides, the tabu mechanism of TS is adopted in this phase to avoid premature convergence. Further details of the metaheuristic are provided in Section V.

In the task scheduling phase, a set of task scheduling subproblems on every single orbit is separately solved by a B&B method based on the allocation results in the task allocation phase. The scheduling scheme of an orbit is exactly regarded as the solution of a subproblem, and the overall scheduling scheme can be obtained by merging all the subscheduling schemes.

The final scheduling results are obtained through iteratively performing the task allocation phase and task scheduling phase until the algorithm termination conditions are met. These two phases are interconnected via the pheromone mechanism and tabu mechanism. This framework can effectively reduce the scheduling complexity of the original problem and obtain a more promising solution by merging a set of subsolutions obtained by a B&B method that solves a set of relatively small-scale subproblems.

IV. MATHEMATICAL MODEL FOR TASK SCHEDULING ON AN ORBIT

In this section, we develop a task scheduling model for a single orbit. Satellite observation operations in practical applications are affected by various factors, such as cloud coverage, imaging data transmission, and satellite malfunction [3], [43]. For the convenience of modeling, we assume that the impacts of these real-world factors are ignored. Besides, we assume that each task is desired to be observed once, without repeated observation requests. The task scheduling model aims to maximize the overall profits of all the scheduled tasks, while satisfying the constraints related to satellite operations, including satellite transfer time between two consecutive tasks, energy capacity, and memory capacity. The profit of a task represents the importance and value to the user of completing the observation task [22], [33], [38].

The used notations are summarized in Table I. Let $O = \{O_1, O_2, \ldots, O_H\}$ be the set of orbits within the predefined scheduling horizon and H the number of orbits. Denote $T = \{1, 2, \ldots, N\}$ as the set of tasks and N the number of tasks. Define $T_k \in T$ as the set of tasks allocated to the orbit $k \in O$.

TABLE I
Notations for the Scheduling Model

Notations	Description
O	Set of orbits
H	Number of orbits
T	Set of tasks
N	Number of tasks
T_k	Set of tasks on orbit k, $T_k \in T$
x_{ij}^k	Decision variables
φ_i	Profit of task i
M_k, E_k	Memory capacity and energy capacity of orbit k
m_k, e_k	Memory and energy consumption for each unit time of observation of orbit k
$[ws_{ik}, we_{ik}]$	Time window for task i on orbit k
θ_{ik}	Slewing angle for task i on orbit k
st_{ij}^k	Transfer time between task i and task j on orbit k
td_k, tu_k, ts_k	Times of sensor shutdown, startup and attitude stabilization on orbit k
v	Slewing velocity of sensors

Each orbit k is associated with a memory capacity M_k and an energy capacity E_k. The observation activity consumes energy and memory resources on each orbit per unit time. Each task $i \in T_k$ is endowed with an observation profit φ_i, a slewing angle θ_{ik}, and a time window $[ws_{ik}, we_{ik}]$ specified by its earliest possible observation time ws_{ik} and its latest possible observation time we_{ik}.

Satellite transfer time is required to observe two different tasks successively. Specifically, after observing a task $i \in T_k$, the satellite needs a sequence of transformation operations to observe the next task $j \in T_k$, including sensor shutdown, slewing, attitude stability, and startup. Denote tu_k, td_k, and ts_k as the time consumption of sensor startup, shutdown, and attitude stability on the orbit k, respectively. Let v be the slewing velocity of the satellite. The transfer time st_{ij}^k can be computed as

$$st_{ij}^k = td_k + \left|\theta_{ik} - \theta_{jk}\right| / v + ts_k + tu_k. \tag{1}$$

To accomplish the model, we introduce binary decision variables x_{ij}^k, and define $x_{ij}^k = 1$ if the task i is the immediate predecessor of the task j on the orbit k; otherwise $x_{ij}^k = 0$. Note that there exist two dummy tasks used to start or terminate sensors when the task index equals 0 or $N + 1$. The dummy tasks do not have real profit (i.e., $\varphi_0 = 0$, $\varphi_{N+1} = 0$). Thus, the integer programming formulation of

the task scheduling on the single orbit is constructed as

$$\max \sum_{i\in T_k} \sum_{j\in T_k\cup\{N+1\}, j\neq i} x_{ij}^k \cdot \varphi_i \tag{2}$$

$$\sum_{i\in T_k} \sum_{j\in T_k\cup\{N+1\}, j\neq i} x_{ij}^k \cdot e_i \cdot (we_{ik} - ws_{ik}) \leq E_k \tag{3}$$

$$\sum_{i\in T_k} \sum_{j\in T_k\cup\{N+1\}, j\neq i} x_{ij}^k \cdot m_k \cdot (we_{ik} - ws_{ik}) \leq M_k \tag{4}$$

$$x_{ij}^k \cdot (ws_{jk} - we_{ik} - st_{ij}^k) \geq 0 \forall i, j \in T_k \tag{5}$$

$$\sum_{j\in T_k\cup\{0\}, j\neq i} x_{ji}^k \leq 1 \,\forall i \in T_k \tag{6}$$

$$\sum_{j\in T_k\cup\{N+1\}, j\neq i} x_{ij}^k \leq 1 \,\forall i \in T_k \tag{7}$$

$$\sum_{j\in T_k\cup\{0\}, j\neq i} x_{ji}^k - \sum_{j\in T_k\cup\{N+1\}, j\neq i} x_{ij}^k = 0 \,\forall i \in T_k \tag{8}$$

$$x_{0,N+1}^k + x_{N+1,0}^k = 0 \tag{9}$$

$$\sum_{j\in T_k} x_{0,j}^k = 1 \tag{10}$$

$$\sum_{i\in T_k} x_{i,N+1}^k = 1. \tag{11}$$

The objective function (2) aims to maximize the entire observation profits of all the scheduled tasks. Constraints (3)–(5) represent the energy, memory, and time window constraints, respectively. Constraints (6) and (7) indicate that there is at most one predecessor task and one subsequent task for each real task. Constraint (8) guarantees that the number of predecessors is equal to the number of successors for each task. Constraint (9) indicates that the dummy tasks cannot be used as adjacent tasks. Since adjacent dummy tasks do not contribute to the objective function, we force $x_{0,N+1}^k$=0 and $x_{N+1,0}^k = 0$. Constraints (10) and (11) ensure that there must be a real task after the dummy task 0 and a real task before the dummy task $N + 1$.

V. ENSEMBLE OF METAHEURISTIC AND EXACT ALGORITHM

A. Metaheuristic for Task Allocation

To cooperate with the proposed DCF, we propose a novel metaheuristic method hybridizing the tabu mechanism of TS and the pheromone mechanism of ACO, for the task allocation phase. In detail, the tabu mechanism is utilized to modify the orbit set for task allocation, while the pheromone mechanism is adopted to select appropriate orbits for tasks. Furthermore, we introduce three kinds of factors, including status factor, pheromone trail factor, and feedback factors to implement the aforementioned two mechanisms.

Before detailing the proposed metaheuristic method, we clarify some definitions for convenience. Denote CO_i as an orbit set in which all orbits have visible time windows for the task i. Although all the orbits in CO_i are visible to i, some of them would not be used due to the tabu mechanism. Thus, denote $CO_i'(n) \subseteq CO_i$ as the available orbits at the nth iteration. Let $w_i(n)$ be the allocation priority of a task i at the nth iteration. Denote $\sigma_{ik}(n) \in [0, l]$ as the tabu factor between a task i and an orbit k at the nth iteration, where l is the tabu step. $\sigma_{ik}(n)$ is used to check whether an orbit k can be utilized in $CO_i'(n)$ when allocating i. Furthermore, define $\eta_{ik}(n)$ and $\tau_{ik}(n)$ as the status factor and the pheromone trail factor between a task i and an orbit k, respectively. $\eta_{ik}(n)$ represents the conflict and load status of a task i on an orbit k. $\eta_{ik}(n)$ and $\tau_{ik}(n)$ are used to calculate the allocation probability $p_{ik}(n)$ of assigning a task i to an orbit k at the nth iteration, which can be expressed by

$$p_{ik}(n) = \frac{[\tau_{ik}(n)]^\alpha \times \left[\eta_{ik}(n)\right]^\beta}{\sum_{t\in CO_i'(n)} [\tau_{it}(n)]^\alpha \times \left[\eta_{it}(n)\right]^\beta} \tag{12}$$

where parameters α and β are real numbers that determine the relative influence of the pheromone trail and the status information.

An overview of the metaheuristic for task allocation is illustrated in Fig. 3. When performing the task allocation, tasks in the task set T are first sorted in descending order according to their allocation priority $w_i(n)$. Then, an available orbit set $CO_i'(n)$ is selected for the task i with the highest allocation priority. Afterward, the allocation probabilities between the task i and each orbit in $CO_i'(n)$ are calculated, and i is allocated to an orbit k according to the allocation probability $p_{ik}(n)$. Meanwhile, the status factor is updated and the assigned task i is removed from T. The aforementioned processes are repeated iteratively until T is empty. After each iteration, the pheromone trail factor and feedback factors will be updated according to the scheduling results. The results of task allocation after each iteration are different, which can make the algorithm escape from local optima and converge to better solutions gradually.

In particular, the status factor is designed based on the task load condition and task conflict condition, and it will be updated once a task is allocated. The pheromone trail factor is updated at each iteration, during which it will be increased, decayed, and diluted according to the scheduling results. The calculation methods of the status factor, pheromone trail factor, and feedback factors are presented as follows:

1) *Status Factor:* Denote T_k' as a set of tasks already allocated to the orbit k, the load degree of k can be expressed by $\varepsilon_k = |T_k'|$ (i.e., the number of tasks already allocated to k). When allocating a new task i, the orbit k in $CO_i'(n)$ presents one of the following two states:

1) there are no allocated tasks on the orbit k (i.e., $\varepsilon_k = 0$);
2) there are some tasks already allocated to the orbit k (i.e., $\varepsilon_k > 0$).

In the second state, the task i to be assigned may conflict with other tasks in T_k'. We use a binary variable $\text{conf}_{is}^k \in \{0, 1\}$ to measure the conflict between the tasks i and $s \in T_k'$. If $\text{conf}_{ij}^k = 1$, the task i conflicts with task j; 0 otherwise. Two tasks are in conflict when their observation

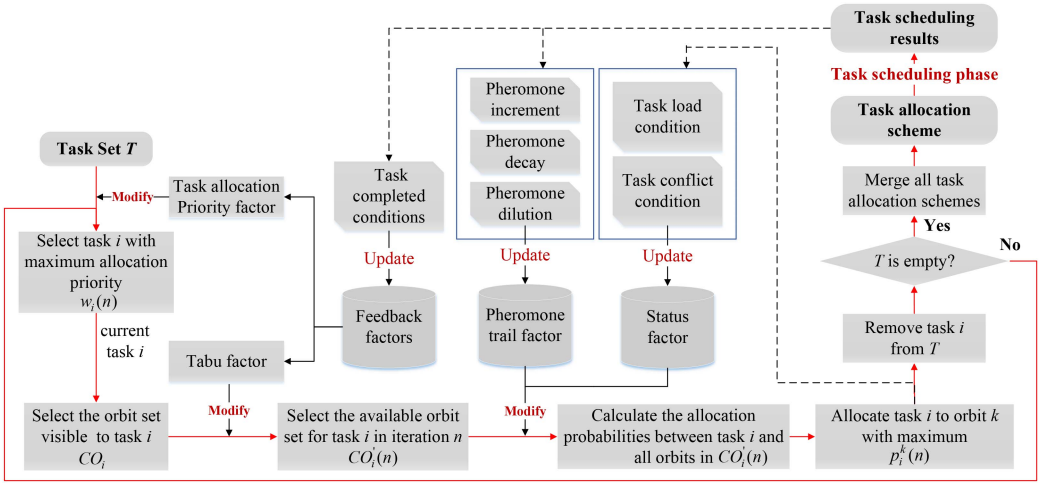

Fig. 3. Task allocation process.

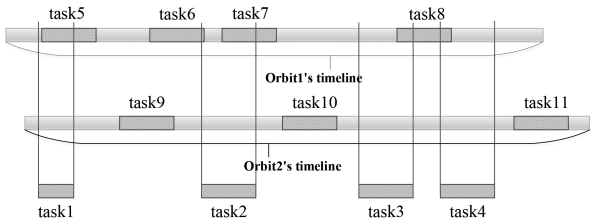

Fig. 4. Conflict and load condition.

time windows satisfy the following conditions:

$$\begin{cases} we_{ik} + |\theta_{ik} - \theta_{jk}|/v > ws_{jk}, & \text{if } ws_{jk} > ws_{ik} \\ we_{jk} + |\theta_{ik} - \theta_{jk}|/v > ws_{ik}, & \text{if } ws_{jk} \le ws_{ik}. \end{cases} \tag{13}$$

Then, let ξ_{ik} be the conflict degree between task i and all the tasks in T_k', it can be calculated according to

$$\xi_{ik} = \sum_{s \in T_k'} \text{conf}_{is}^k. \tag{14}$$

Fig. 4 shows an example of the conflict degree and load degree, in which $\xi_{11} = 1, \xi_{12} = 0, \xi_{21} = 2, \xi_{22} = 0, \xi_{31} = 1, \xi_{32} = 0, \xi_{41} = 1, \xi_{42} = 0, \varepsilon_1 = 4$, and $\varepsilon_2 = 3$.

Denote $\bar{\varepsilon}_k$ and $\bar{\xi}_{ik}$ as the normalized values of ε_k and ξ_{ik}, respectively, they can be calculated by

$$\bar{\varepsilon}_k = \left(\sum_{s \in CO_i(n)} \varepsilon_s - \varepsilon_k \right) \Big/ \sum_{s \in CO_i(n)} \varepsilon_s \tag{15}$$

$$\bar{\xi}_{ik} = \left(\sum_{s \in CO_i(n)} \xi_{is} - \xi_{ik} \right) \Big/ \sum_{s \in CO_i(n)} \xi_{is}. \tag{16}$$

Therefore, the value of the status factor $\eta_{ik}(n)$ can be obtained by

$$\eta_{ik}(n) = a \cdot \bar{\varepsilon}_k + b \cdot \bar{\xi}_{ik} \tag{17}$$

where a and b are the weights of $\bar{\varepsilon}_k$ and $\bar{\xi}_{ik}$, respectively.

2) *Pheromone Trail Factor:* We use $\rho \in (0, 1)$ and λ to represent the pheromone decay parameter and pheromone dilution parameter, respectively. The increment of the pheromone between a task i and an orbit k can be expressed as

$$\Delta\tau_{ik} = \frac{\gamma(n)}{\lambda \cdot \text{Num}(n)} \tag{18}$$

where $\gamma(n)$ and $\text{Num}(n)$ are scheduling profits and number of scheduled tasks at the nth iteration, respectively. Denote γ^* as the current best scheduling profits, it can be calculated as

$$\gamma^* = \max\{\gamma(1), \gamma(2), \ldots, \gamma(n)\}. \tag{19}$$

When $\sum_{j \in T_k \cup \{N+1\}, j \neq i} x_{ij}^k = 1$, which indicates that the task i is successfully scheduled, the pheromone trail factor $\tau_{ik}(n)$ would be updated based on the following formulas:

$$\begin{cases} \tau_{ik}(1+n) = (1-\rho) \cdot \tau_{ik}(n) + \Delta\tau_{ik} \\ \tau_{ik}(1) = 1. \end{cases} \tag{20}$$

3) *Feedback Factors:* The feedback factors consist of the allocation priority factor $w_i(n)$ and tabu factor $\sigma_{ik}(n)$. After the task scheduling on each orbit has been finished, we use $w_i(n)$ and $\sigma_{ik}(n)$ to update the allocation orders of the task i, as well as the elements in $CO_i'(n)$ at the $(n+1)$th iteration, which will affect the task allocation process at the next iteration.

The initial value of the allocation priority factor is the initial profit of the task i (i.e., $w_i(1) = \varphi_i$). If the task i has not been successfully scheduled (i.e., $\sum_{j \in T_k \cup \{N+1\}, j \neq i} x_{ij}^k = 0$), $w_i(n+1)$ will be updated; otherwise it keeps unchanged, which can be expressed as

$$\begin{cases} w_i(n+1) = w_i(n) - c, & \text{if } \sum_{j \in T_k \cup \{N+1\}, j \neq i} x_{ij}^k = 0 \\ w_i(n+1) = w_i(n), & \text{otherwise} \end{cases} \tag{21}$$

where $c \in \mathbb{Z}_+$ is a weight decay parameter. The allocation priority of a task will be decreased if the task is not scheduled, thereby ensuring that the task that has not been scheduled for several times will be adjusted backward in the task allocation sequence.

The tabu factor $\sigma_{ik}(n)$ checks whether an orbit is available or not. The initial value of $\sigma_{ik}(n)$ is 0, and it will be updated iteratively in the later iterations. We propose seven heuristic rules to determine the value of $\sigma_{ik}(n)$. Specifically, the initialization of the tabu process is shown in *Rule 1*. If $\sigma_{ik}(n) > 0$, which indicates that the orbit k is not allowed to serve the task i at the nth iteration, the elements in $CO'_i(n)$ will be adjusted according to *Rules 2–4*. If $\sigma_{ik} = 0$, which means that the orbit k is available to serve the task i at the nth iteration, $CO'_i(n)$ will be updated based on *Rule 5*. Finally, the tabu factor $\sigma_{ik}(n)$ will be updated iteratively according to *Rules 6 and 7*.

Rule1: In the initial situation, all the orbits in CO_i can be used to serve the task i, which can be expressed as $CO'_i(1) = CO_i$.

Rule2: If there is only one visible orbit for the task i (i.e., $|CO_i| = 1$), the orbit in CO_i is always available, which can be expressed by $CO'_i(n) \equiv CO_i$.

Rule3: If there are multiple visible orbits for the task i (i.e., $|CO_i| > 1$), a randomly selected orbit k would be removed from $CO'_i(n)$ to obtain $CO'_i(n+1)$, which is denoted as $CO'_i(n+1) = CO'_i(n)/\{O_k\}$.

Rule4: If there are multiple visible orbits and no orbits can be used for serving the task i at the nth iteration (i.e., $|CO_i| > 1$ and $CO'_i(n) = \emptyset$), $CO'_i(n)$ will be initialized as $CO'_i(n+1) = CO_i$.

Rule5: In the iterative process, when $\sigma_{ik}(n)$ decreases to 0, the orbit k would be added to $CO'_i(n+1)$ in the next iteration, which is written as $CO'_i(n+1) = CO'_i(n) \cup O_k$.

Rule6: If the task i allocated to the orbit k is not successfully scheduled at the nth iteration, the tabu factor $\sigma_{ik}(n)$ will be updated at the $(n+1)$th iteration as $\sigma_{ik}(n+1) = l$, where $l \in \mathbb{Z}_+$.

Rule7: If the orbit k is forbidden to serve the task i at the nth iteration (i.e., $\sigma_{ik}(n) > 0$), the tabu factor $\sigma_{ik}(n)$ will decrease according to $\sigma_{ik}(n+1) = \sigma_{ik}(n) - \Delta l$ in the following iterations until its value is 0. Here, Δl is a divisor of l, which is used to gradually reduce the value of $\sigma_{ik}(n)$.

B. Ensemble of Metaheuristic and Exact Algorithm Based on the DCF

The pseudocode of the EHE-DCF is provided in Algorithm 1. In the algorithm, first, the task i with the highest allocation priority is selected as the current task to be allocated (line 5). Second, according to the visibility of the orbits to the task i and the orbit tabu condition, an available orbit set $CO'_i(n)$ is derived for serving task i (line 6). Third, the allocation probabilities between the task i and each orbit in $CO'_i(n)$ are calculated based on the pheromone trail factor $\tau_{ik}(n)$ and status factor $\eta_{ik}(n)$, and the task i is allocated to the orbit k based on the allocation probability $p_{ik}(n)$ (lines 8 and 9). When all the tasks in T are scheduled, the task allocation phase is terminated. In the single orbit scheduling phase, we use the CPLEX software to implement the B&B method to obtain a set of scheduling results of the single orbit scheduling problems. Then, the subscheduling results are merged to obtain an overall task scheduling result (lines 13 and 14). The task allocation and the single orbit scheduling phases are performed iteratively until the number of iterations reaches the maximum iterations $G1$. Specifically, the status factor $\eta_{ik}(n)$ is updated once the task i is assigned to the orbit k (line 11), and the pheromone pheromone trail factor $\tau_{ik}(n)$ and feedback factors (i.e., allocation priority factor $w_i(n)$ and tabu factor $\sigma_{ik}(n)$) are updated after each iteration (line 16).

Algorithm 1: EHE-DCF.

Input: Task set T, orbit set O, time windows between tasks and orbits, maximum iterations G
Output: scheduling results

```
1  Initialize algorithm parameters
2  while n ≤ G1 do
3      //Task allocation phase
4      while T ≠ ∅ do
5          Find the task i that has the highest allocation priority
6          Select an available orbit set CO'_i(n) for the task i
7          for each orbit k ∈ CO'_i(n) do
8              Calculate the allocation probability p_ik(n) according to τ_ik(n) and η_ik(n)
9          Allocate the task i to the orbit k according to p_ik(n)
10         Remove the task i from T
11         Update status factor η_ik(n)
12     //Task scheduling phase
13     for each orbit k scheduled do
14         Solve the scheduling problem on orbit k by B&B method
15     Merge all the sub-scheduling results
16     Update pheromone trail factor τ_ik(n), and feedback factors w_i(n) and σ_ik(n).
17     n = n + 1
```

C. Complexity Analysis

The proposed EHE-DCF repeats the task allocation phase and the task scheduling phase with G iterations. In the task allocation phase, assume that N tasks are allocated to H orbits and the number of tasks on each orbit is uncertain. The metaheuristic developed for the task allocation has a computational complexity $O(N \cdot (\log(N) + H))$ by using a quick sort algorithm [44]. Then, in the task scheduling phase, we denote the highest number of tasks allocated to a single orbit is $N_{\max} \in [0, N]$. Since the single orbit scheduling can be regarded as a knapsack problem [45], the complexity of the single orbit scheduling is expressed as $O(2^{N^2_{\max}})$ by assuming a B&B based full-factorial search. The task allocation phase, which includes H single scheduling problems, thus

has a complexity $O(2^{N_{\max}^2} \cdot H)$. Therefore, the complexity of the EHE-DCF is $O(G \cdot (N \cdot \log(N) + N \cdot H + 2^{N_{max}^2} \cdot H))$.

VI. SIMULATION EXPERIMENTS

In this section, we carry out experiments based on EOS scheduling instances with different task scales and different observation resource scales to comprehensively evaluate the performance of the EHE-DCF. Both EHE-DCF and its competitors are implemented in MATLAB R2016a and CPLEX12.5, and all the experiments are executed on a computer with Intel(R) Core (TM) i5 2.80 GHz and 8.0-GB RAM.

A. Comparative Algorithms

We compare the EHE-DCF with five algorithms, including three DCF-based metaheuristics, a state-of-the-art metaheuristic in the existing literature, and pure B&B method. The pure B&B method is implemented with the commercial solver CPLEX, and the metaheuristics are briefly introduced as follows:

1) *Greedy Algorithm Based on the DCF (GR-DCF):* Greedy algorithms that preferentially schedule the task with the highest profit or priority are commonly used to solve satellite scheduling problems in practical applications [46], [47]. In the GR-DCF, the task allocation phase is the same as EHE-DCF, while the task scheduling phase is performed greedily. In each iteration of the task scheduling phase, the tasks assigned to each orbit are scheduled iteratively according to their profits. During the scheduling process, if constraints (3) and (4) are violated when a task is inserted into an orbit, the previously scheduled tasks on this orbit would be removed one by one, until all constraints are satisfied. The scheduled task with the lowest profit would be removed first. Finally, the tasks that are not successfully scheduled are preserved for the next iteration.

2) *Simulated Annealing Neighborhood Based on Greedy Neighborhood Structure and DCF (SANS1-DCF):* Different from the GR-DCF, the SANS1-DCF implements a greedy neighborhood structure in the task scheduling phase. The greedy neighborhood structure schedules tasks assigned to each orbit in the same way as GR-DCF, but removes a task with the lowest profit before inserting tasks if any task has been inserted into the orbit. By removing tasks from previously scheduled results, The SANS1-DCF is expected to have a higher capability to escape from the local optimum. Meanwhile, the SANS1-DCF adopts the well-known Metropolis acceptance criteria [48] of the simulated annealing algorithm to accept worse solutions with an adaptively controlled probability.

3) *Simulated Annealing Neighborhood Based on Random Neighborhood Structure and DCF (SANS2-DCF):* The framework of the SANS2-DCF is similar to SANS1-DCF. The difference between SANS1-DCF and SANS2-DCF is that SANS2-DCF removes a task randomly instead of removing the task with the lowest profit from the previously scheduled result.

4) *Adaptive Simulated Annealing-Based Scheduling Algorithm (ASA):* The ASA is a state-of-the-art algorithm extracted from the existing literature [19]. This algorithm has been proved efficient in solving EOS scheduling problems, due to involving sophisticated mechanisms, i.e., adaptive temperature control, tabu-list-based short-term revisiting avoidance mechanism, and intelligent combination of neighborhood structures.

B. Experiment Setups

In the experimental studies, eight instances varying from 200 to 1600 tasks are prepared. The ground targets are distributed in a range of latitude 15°–45° and longitude 80°–120° randomly. The profits of tasks are random values within [1, 10]. We set the allowable runtime for an algorithm solving a scheduling problem to 3 600 s and the scheduling horizon to 24 h. The basic information of the instances are provided in Table II. We consider ten EOSs and each satellite is equipped with a sensor to accomplish observation tasks. Detailed satellite orbital parameters for simulations are displayed in Table III. Parameters of satellite and algorithms are listed in Tables IV and V, respectively. All algorithms are repeated 25 times on each instance independently.

C. Results and Discussions

The results are reported in Table VI, including the obtained observation profits, number of scheduled tasks, and average runtime. As seen from Table VI, the EHE-DCF outperforms its comparative metaheuristics (i.e., GR-DCF, SANS1-DCF, SANS2-DCF, and ASA) in terms of the obtained observation profits and the number of scheduled tasks. This is because EHE-DCF uses the B&B method to generate optimal solutions for single orbit scheduling subproblems and the iterative task allocation procedure realizes a proper problem partition. Meanwhile, these two phases can work cooperatively to obtain a high-quality entire scheduling scheme.

By contrast, pure B&B can get the highest profits in the small-scale task scheduling problems without surprise, as pure B&B is an exact algorithm. However, its runtime increases dramatically when the task scale increases, indicating that its computational efficiency is not satisfactory when solving large-scale EOS scheduling problems. To be concrete, pure B&B has a sharp increase in runtime when the number of tasks is more than 600, and it consumes much more time than other comparative algorithms. In the case of 800 tasks, the runtime of pure B&B has exceeded the predefined allowable running time. As for the ASA, it needs more computational efforts to solve the scheduling problem, while the obtained profits are less than that of other metaheuristics based on the DCF. Particularly, the ASA cannot solve large-scale instances (i.e., instances C5–C8) within acceptable running time. Although pure B&B consumes less computational time when solving small-scale instances (i.e., instances C1–C3) compared with the ASA, the pure B&B still requires more

TABLE II
Information of Instances

Scheduling horizon	Number of satellites	Number of tasks	Profit of task	Acceptable runtime
24 hours	10	[200, 1600]	[1,10]	3600 seconds

TABLE III
Satellite Orbital Parameters

Orbital parameters	Values
Semimajor axis (km)	6678.14
Inclination (°)	8.5, 28.5, 48.5, 68.5, 88.5, 108.5, 128.5, 148.5, 158.5, 168.5
Right ascension of the ascending node (°)	0
Eccentricity	0
Argument of perigee (°)	0
Mean anomaly (°)	0

TABLE IV
Parameters of Satellites

E_k	M_k	e_k	m_k	td_k	ts_k	tu_k	v
300	2400	1	1	5	3	5	1

TABLE V
Parameters of Algorithms

Algorithms	Parameters
EHE-DCF	$\alpha = 3$, $\beta = 3$, $a = 0.7$, $b = 0.3$, $c = 0.2$, $l = 2$, $\Delta l = 1$, $\rho = 0.1$, $G = 200$.
GR-DCF SANS1-DCF SANS2-DCF	Maximum iterations $G2 = 500$, start temperature $Tem_s = 300$, end temperature $Tem_e = 0.001$, cooling rate $\delta = 0.99$.
ASA	Maximum iterations $G3 = 5 * N$ and the other parameters are the same as in [19].

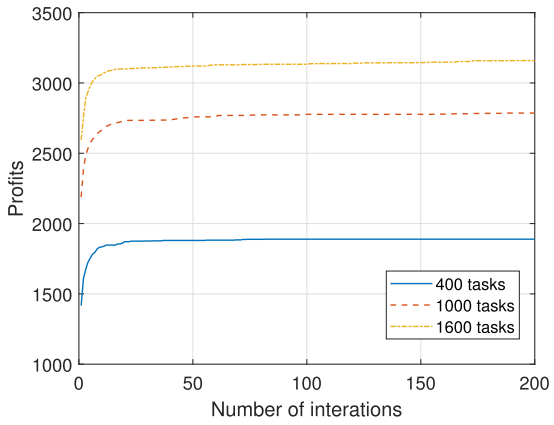

Fig. 5. Convergence of the EHE-DCF on the instances with 400, 1000, and 1600 tasks.

computational time compared with DCF-based metaheuristics on instances C2–C8. These observation results further prove the efficiency of the DCF, particularly on large-scale instances.

To visually analysis the performance of the EHE-DCF, we plot the convergence process of the EHE-DCF when dealing with the instances with 400, 1000, and 1600 tasks in Fig. 5, which demonstrates that the EHE-DCF is robust and able to converge to a satisfactory solution efficiently.

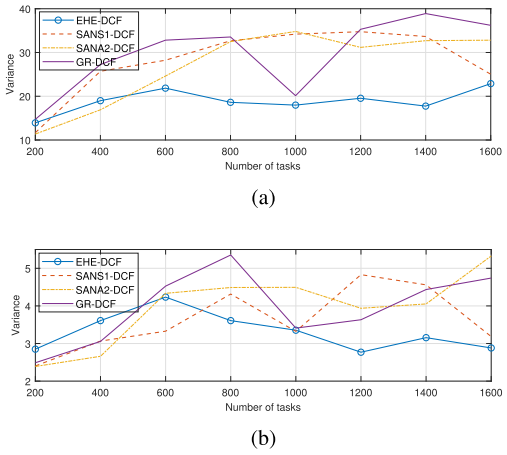

Fig. 6. Comparisons on the stability with respect to the obtained profits and the number of scheduled tasks. (a) Variance of the obtained profits. (b) Variance of the number of scheduled tasks.

The variances of the scheduling profits and number of scheduled tasks obtained by four DCF-based metaheuristics on all instances are investigated in Fig. 6. It can be found that the EHE-DCF has more stable performance in solving EOS scheduling problems with different scales compared with SANS1-DCF, SANS2-DCF, and GR-DCF. In particular, its advantage is more obvious when the task scale is more than 800, which indicates that EHE-DCF could be useful and reliable in practical applications.

To further demonstrate the advantages of the EHE-DCF under large-scale task scheduling, we define two indicators, i.e., the increase rate of obtained profits $r_{1,i}^{\text{profit}}$ and the number of scheduled tasks $r_{1,i}^{\text{task}}$, which are, respectively, calculated as follows:

$$r_{1,i}^{\text{profit}} = \frac{\gamma_1^{\text{ave}} - \gamma_i^{\text{ave}}}{\gamma_i^{\text{ave}}} \times 100\%, i = \{2, 3, 4\} \tag{22}$$

$$r_{1,i}^{\text{task}} = \frac{\text{Num}_1^{\text{ave}} - \text{Num}_i^{\text{ave}}}{\text{Num}_i^{\text{ave}}} \times 100\%, i = \{2, 3, 4\} \tag{23}$$

where γ_i^{ave} and $\text{Num}_i^{\text{ave}}$ represent the average value of obtained profits and the number of scheduled tasks of the DCF-based algorithm i, respectively. The comparison results of increase rates are shown in Fig. 7. The results show that when the task scale is 1600, the increase rates of the obtained profits and number of scheduled tasks reach the maximum values. It can be concluded that the performance of three comparative DCF-based metaheuristics (i.e., GR-DCF, SANS1-DCF, and SANS2-DCF) are deteriorated when solving large-scale EOS scheduling problems. On the contrary, the increase rates of the EHE-DCF increase significantly in the case

TABLE VI
Experimental Results on Eight Instances

Instance	Task scale	Algorithm	Profits			Number of scheduled tasks			Runtime (s)
			Min.	Ave.	Max.	Min.	Ave.	Max.	
C1	200	EHE-DCF	1063	1095.84	1117	160	166	170	16.131
		SANS1-DCF	1046	1073.88	1091	157	162	167	10.297
		SANS2-DCF	1054	1077.88	1102	157	163	167	9.056
		GR-DCF	1036	1060.52	1084	157	162	168	4.734
		Pure B&B	\	1198.00	\	\	186	\	3.104
		ASA	1050	1069.90	1098	161	164	168	37.866
C2	400	EHE-DCF	1833	1859.48	1902	253	259	266	27.744
		SANS1-DCF	1750	1781.28	1848	242	247	253	44.782
		SANS2-DCF	1773	1807.12	1834	242	249	252	41.367
		GR-DCF	1654	1728.92	1781	237	246	250	18.488
		Pure B&B	\	2134.00	\	\	306	\	240.753
		ASA	1541	1591.00	1653	215	223	231	432.491
C3	600	EHE-DCF	2235	2280.64	2312	293	302	308	54.801
		SANS1-DCF	2078	2133.64	2181	273	280	287	153.926
		SANS2-DCF	2149	2208.36	2252	277	287	297	147.857
		GR-DCF	1974	2042.08	2100	270	277	287	62.184
		Pure B&B	\	2616.00	\	\	350	\	448.514
		ASA	1703	1744.00	1801	231	237	243	1583.542
C4	800	EHE-DCF	2531	2577.92	2616	318	327	335	102.31
		SANS1-DCF	2306	2358.92	2420	287	296	305	383.494
		SANS2-DCF	2407	2465.12	2521	298	307	314	376.98
		GR-DCF	2194	2255.76	2316	286	295	309	159.789
		Pure B&B	\	2928.00	\	\	371	\	4472.515
		ASA	1877	1923.90	1976	238	247	255	3519.338
C5	1000	EHE-DCF	2762	2796.36	2829	341	346	354	170.073
		SANS1-DCF	2430	2530.28	2588	303	310	315	756.406
		SANS2-DCF	2544	2621.84	2674	307	317	324	767.229
		GR-DCF	2371	2403.04	2467	301	307	313	322.98
C6	1200	EHE-DCF	2903	2936.04	2976	354	358	365	274.739
		SANS1-DCF	2576	2628.68	2684	306	315	326	1357.924
		SANS2-DCF	2729	2769.96	2857	323	329	338	1338.86
		GR-DCF	2407	2483.60	2562	306	313	320	574.383
C7	1400	EHE-DCF	3036	3074.88	3108	361	368	372	422.069
		SANS1-DCF	2676	2731.80	2795	314	322	331	2178.727
		SANS2-DCF	2818	2883.56	2943	328	337	344	2183.903
		GR-DCF	2509	2577.417	2680	310	319	330	922.881
C8	1600	EHE-DCF	3108	3149.08	3187	368	374	379	620.367
		SANS1-DCF	2738	2789.16	2831	320	326	333	3354.379
		SANS2-DCF	2883	2945.48	3006	330	341	350	3318.207
		GR-DCF	2555	2634.80	2693	314	322	333	1396.34

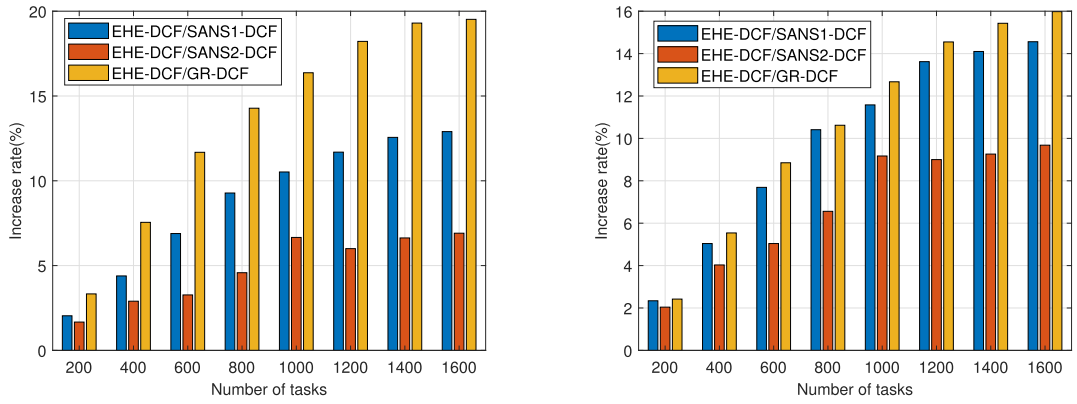

Fig. 7. Histograms on different task scales with respect to the obtained profits and the number of scheduled tasks. (a) Increase rate of the obtained profits. (b) Increase rate of the number of scheduled tasks.

of large-scale tasks, indicating that the EHE-DCF is particularly suitable to large-scale complex EOS scheduling problems.

More observation resources mean more observation opportunities, while it brings more scheduling challenges. To test the performance of the EHE-DCF in solving EOS scheduling problems with different numbers of observation resources, we apply EHE-DCF, GR-DCF, SANS1-DCF, and SANS2-DCF to instances with a different number of satellites and the same number of tasks. Five groups of instances are implemented by setting the number of satellites from 2 to 10 and the number of tasks to 1000. The ASA is not

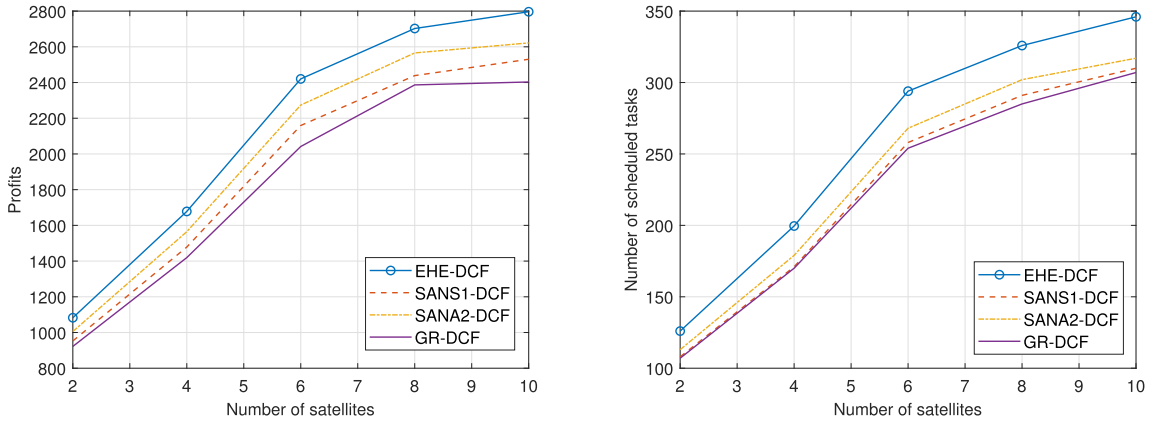

Fig. 8. Comparisons on different satellite scales with respect to the obtained profits and the number of scheduled tasks. (a) Profits obtained by satellites with different scales. (b) Number of tasks scheduled by satellites with different scales.

tested here for comparisons, as we have shown the superior performance of the DCF-based metaheuristics in Table VI. The computational results based on the EHE-DCF, SANS1-DCF, SANS2-DCF, and GR-DCF are plotted in Fig. 8. It can be seen that the number of satellites shows a significant effect on the results of scheduling. Meanwhile, the scheduling results obtained by the EHE-DCF are significantly better than those of other comparative algorithms on all instances.

VII. CONCLUSION

In this article, a novel ensemble approach named ECE-DCF, which combines metaheuristic and exact methods based on a DCF, has been proposed for solving the multiple EOS scheduling problem. The ECE-DCF divides the EOS scheduling problem into a task allocation phase and a single orbit scheduling phase. In the task allocation phase, a metaheuristic is designed to generate a fairly reasonable task allocation scheme in an iterative manner. This metaheuristic method involves sophisticated mechanisms, i.e., probabilistic selection and tabu mechanism, feedback factors, status factor, and pheromone trail factors. In the single orbit scheduling phase, we construct an integer programming model and adopt the B&B method to obtain an optimal solution for each subproblem. Furthermore, these two phases are performed iteratively and interactively to solve the EOS scheduling problem. Compared with an exact method (i.e., pure B&B), three DCF-based metaheuristic (i.e., GR-DCF, SANS1-DCF, and SANS2-DCF), and a state-of-the-art metaheuristic (i.e., ASA), the -DCF outperforms the competitors in terms of scheduling profits and number of scheduled tasks on the most instances, as well as running time on large-scale instances. Extensive experiments are further conducted to demonstrate that the EHE-DCF is a robust and efficient method for solving EOS scheduling problems, especially when the scale of the scheduling problem gets large. In future studies, we would extend the proposed approach to solve more complicated EOS scheduling problems, such as the agile EOS scheduling problem [49].

REFERENCES

[1] G. Wu, W. Pedrycz, H. Li, M. Ma, and J. Liu
Coordinated planning of heterogeneous earth observation resources
IEEE Trans. Syst., Man, Cybern., Syst., vol. 46, no. 1, pp. 109–125, Jan. 2016.

[2] J. Li, C. Li, and F. Wang
Automatic scheduling for Earth observation satellite with temporal specifications
IEEE Trans. Aerosp. Electron. Syst., vol. 56, no. 4, pp. 3162–3169, Aug. 2020.

[3] X. Wang, G. Song, R. Leus, and C. Han
Robust earth observation satellite scheduling with uncertainty of cloud coverage
IEEE Trans. Aerosp. Electron. Syst., vol. 56, no. 3, pp. 2450–2461, Jun. 2020.

[4] T. Stollenwerk, V. Michaud, E. Lobe, M. Picard, A. Basermann, and T. Botter
Agile earth observation satellite scheduling with a quantum annealer
IEEE Trans. Aerosp. Electron. Syst., vol. 57, no. 5, pp. 3520-3528, Oct. 2021.

[5] X. Wang, C. Han, P. Yang, and X. Sun
Onboard satellite visibility prediction using metamodeling based framework
Aerosp. Sci. Technol., vol. 94, 2019, Art. no. 105377.

[6] G. Peng, G. Song, L. Xing, A. Gunawan, and P. Vansteenwegen
An exact algorithm for agile earth observation satellite scheduling with time-dependent profits
Comput. Operations Res., vol. 120, 2020, Art. no. 104946.

[7] V. Gabrel and D. Vanderpooten
Enumeration and interactive selection of efficient paths in a multiple criteria graph for scheduling an earth observing satellite
Eur. J. Oper. Res., vol. 139, no. 3, pp. 533–542, 2002.

[8] K. Zhu, J. Li, and H. Baoyin
Satellite scheduling considering maximum observation coverage time and minimum orbital transfer fuel cost
Acta Astronautica, vol. 66, no. 1, pp. 220–229, 2010.

[9] D. Liao and Y. 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.

[10] Y. Huang, Z. Mu, S. Wu, B. Cui, and Y. Duan
Revising the observation satellite scheduling problem based on deep reinforcement learning
Remote Sens., vol. 13, no. 12, 2021, Art. no. 2377.

[11] X. 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.

[12] J. Wang, E. Demeulemeester, X. Hu, D. Qiu, and J. Liu
Exact and heuristic scheduling algorithms for multiple earth observation satellites under uncertainties of clouds
IEEE Syst. J., vol. 13, no. 3, pp. 3556–3567, Sep. 2019.

[13] H. Chen, S. Yang, J. Li, and N. Jing
Exact and heuristic methods for observing task-oriented satellite cluster agent team formation
Math. Problems Eng., vol. 2018, pp. 1–23, 2018.

[14] J. Zhang, L. Xing, G. Peng, F. Yao, and C. Chen
A large-scale multiobjective satellite data transmission scheduling algorithm based on SVM NSGA-II
Swarm Evol. Comput., vol. 50, 2019, Art. no. 100560.

[15] M. Chen, J. Wen, Y. Song, L. Xing, and Y. Chen
A population perturbation and elimination strategy based genetic algorithm for multi-satellite TT&C scheduling problem
Swarm Evol. Comput., vol. 65, 2021, Art. no. 100912.

[16] Y. Du, L. Xing, J. Zhang, Y. Chen, and Y. He
MOEA based memetic algorithms for multi-objective satellite range scheduling problem
Swarm Evol. Computation, vol. 50, 2019, Art. no. 100576.

[17] W. Zhu, X. Hu, W. Xia, and P. Jin
A two-phase genetic annealing method for integrated earth observation satellite scheduling problems
Soft Comput., vol. 23, no. 1, pp. 181–196, 2019.

[18] J. Abel
A divide and conquer approach to least-squares estimation
IEEE Trans. Aerosp. Electron. Syst., vol. 26, no. 2, pp. 423–427, Mar. 1990.

[19] G. Wu, H. Wang, W. Pedrycz, H. Li, and L. Wang
Satellite observation scheduling with a novel adaptive simulated annealing algorithm and a dynamic task clustering strategy
Comput. Ind. Eng., vol. 113, pp. 576–588, 2017.

[20] X. Wang, R. Leus, and C. Han
Fixed interval scheduling of multiple earth observation satellites with multiple observations
in *Proc. 9th Int. Conf. Mech. Aerosp. Eng.*, 2018, pp. 28–33.

[21] Z. Li and X. Li
A multi-objective binary-encoding differential evolution algorithm for proactive scheduling of agile earth observation satellites
Adv. Space Res., vol. 63, no. 10, pp. 3258–3269, 2019.

[22] J. Wang, E. Demeulemeester, X. Hu, and G. Wu
Expectation and SAA models and algorithms for scheduling of multiple earth observation satellites under the impact of clouds
IEEE Syst. J., vol. 14, no. 4, pp. 5451–5462, Dec. 2020.

[23] X. Wang, Y. Gu, G. Wu, and J. R. Woodward
Robust scheduling for multiple agile earth observation satellites under cloud coverage uncertainty
Comput. Ind. Eng., vol. 156, 2021, Art. no. 107292.

[24] M. Deng *et al.*
A two-phase coordinated planning approach for heterogeneous earth-observation resources to monitor area targets
IEEE Trans. Syst., Man, Cybern., Syst., vol. 51, no. 10, pp. 6388–6403, Oct. 2021.

[25] B. Sun, W. Wang, and Q. Qi
Satellites scheduling algorithm based on dynamic constraint satisfaction problem
in *Proc. Int. Conf. Comput. Sci. Softw. Eng.*, 2008, vol. 4, pp. 167–170.

[26] C. Plaunt, J. Frank, and K. Jonsson
Satellite tele-communications scheduling as dynamic constraint satisfaction
in *Proc. Artif. Intell., Robot. Automat. Space*, 1999, vol. 440, pp. 277–284.

[27] K. Luo
A hybrid binary artificial bee colony algorithm for the satellite photograph scheduling problem
Eng. Optim., vol. 52, no. 8, pp. 1421–1440, 2020.

[28] M. Vasquez and J.-K. Hao
Upper bounds for the spot 5 daily photograph scheduling problem
J. Combinatorial Optim., vol. 7, no. 1, pp. 87–103, 2003.

[29] H. Wang, Z. Yang, W. Zhou, and D. Li
Online scheduling of image satellites based on neural networks and deep reinforcement learning
Chin. J. Aeronaut., vol. 32, no. 4, pp. 1011–1019, 2019.

[30] P. Wang, X. Zhang, S. Zhang, H. Li, and T. Zhang
Time-expanded graph-based resource allocation over the satellite networks
IEEE Wireless Commun. Lett., vol. 8, no. 2, pp. 360–363, Apr. 2019.

[31] A. Sarkheyli, A. Bagheri, B. Ghorbani-Vaghei, and R. Askari-Moghadam
Using an effective tabu search in interactive resources scheduling problem for LEO satellites missions
Aerosp. Sci. Technol., vol. 29, no. 1, pp. 287–295, 2013.

[32] X. Hu, W. Zhu, B. An, P. Jin, and W. Xia
A branch and price algorithm for EOS constellation imaging and downloading integrated scheduling problem
Comput. Operations Res., vol. 104, pp. 74–89, 2019.

[33] X. Huang, H. Wang, J. Zhu, and M. Ma
Simulation based multi-objective evolutionary algorithm for electronic reconnaissance satellites scheduling problem
in *Proc. 2nd Int. Conf. Power Electron. Intell. Transp. Syst.*, 2009, vol. 1, pp. 166–170.

[34] K. B. Gao, G. H. Wu, and J. H. Zhu
Multi-satellite observation scheduling based on a hybrid ant colony optimization
Adv. Mater. Res., vol. 765, pp. 532–536, 2013.

[35] P. Gao, Y. J. Tan, J. F. Li, and R. J. He
An ant colony algorithm for remote satellite and ground integration scheduling problem in parallel environment
Adv. Mater. Res., vol. 791, pp. 1341–1346, 2013.

[36] 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.

[37] Z. Zhang, F. Hu, and N. Zhang
Ant colony algorithm for satellite control resource scheduling problem
Appl. Intell., vol. 48, no. 10, pp. 3295–3305, 2018.

[38] H. Wang, M. Xu, R. Wang, and Y. Li
Scheduling earth observing satellites with hybrid ant colony optimization algorithm
in *Proc. Int. Conf. Artif. Intell. Comput. Intell.*, 2009, vol. 2, pp. 245–249.

[39] Y. Xu, X. Liu, R. He, and Y. Chen
Multi-satellite scheduling framework and algorithm for very large area observation
Acta Astronautica, vol. 167, pp. 93–107, 2020.

[40] X. Liu, B. Bai, Y. Chen, and F. Yao
Multi satellites scheduling algorithm based on task merging mechanism
Appl. Math. Comput., vol. 230, pp. 687–700, 2014.

[41] G. Wu, M. Ma, J. Zhu, and D. Qiu
Multi-satellite observation integrated scheduling method oriented to emergency tasks and common tasks
J. Syst. Eng. Electron., vol. 23, no. 5, pp. 723–733, 2012.

[42] N. Bianchessi, J. F. Cordeau, J. Desrosiers, G. Laporte, and V. Raymond
A heuristic for the multi-satellite, multi-orbit and multi-user management of earth observation satellites
Eur. J. Oper. Res., vol. 177, no. 2, pp. 750–762, 2007.

[43] G. Wu, Q. Luo, Y. Zhu, X. Chen, Y. Feng, and W. Pedrycz
Flexible task scheduling in data relay satellite networks
IEEE Trans. Aerosp. Electron. Syst., to be published, doi: 10.1109/TAES.2021.3115587.

[44] C. A. Hoare
Quicksort
Comput. J., vol. 5, no. 1, pp. 10–16, 1962.

[45] P. C. Chu and J. E. Beasley
A genetic algorithm for the multidimensional knapsack problem
J. Heuristics, vol. 4, no. 1, pp. 63–86, 1998.

[46] X. Chen, G. Reinelt, G. Dai, and M. Wang
Priority-based and conflict-avoidance heuristics for multi-satellite scheduling
Appl. Soft Comput., vol. 69, pp. 177–191, 2018.

[47] J. Wu, J. Zhang, J. Yang, and L. Xing
Research on task priority model and algorithm for satellite scheduling problem
IEEE Access, vol. 7, pp. 103031–103046, 2019.

[48] P. Gao, W. Li, F. Yao, B. Bai, and J. Yang
Simulated annealing algorithm for EOS scheduling problem with task merging
in *Proc. Int. Conf. Model., Identif. Control*, 2011, pp. 547–552.

[49] X. Wang, G. Wu, L. Xing, and W. Pedrycz
Agile earth observation satellite scheduling over 20 years: Formulations, methods, and future directions
IEEE Syst. J., vol. 15, no. 3, pp. 3881–3892.

Guohua Wu (Member, IEEE) received the B.S. degree in information systems and the Ph.D. degree in operations research from the National University of Defense Technology, Changsha, China, in 2008 and 2014, respectively.

During 2012 and 2014, he was a visiting Ph.D student with the University of Alberta, Edmonton, Canada. He is currently a Professor with the School of Traffic and Transportation Engineering, Central South University, Changsha. His current research interests include planning and scheduling, computational intelligence, and machine learning. He has authored more than 80 referred papers including those published in IEEE TRANSACTIONS ON CYBERNETICS—PART B (IEEE TCYB), IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART A: SYSTEMS AND HUMANS (IEEE TSMCA), and IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS.

Dr. Wu serves as an Associate Editor for *Swarm and Evolutionary Computation Journal*, an editorial board member for *International Journal of Bio-Inspired Computation*, and a Guest Editor for *Information Sciences* and Meme*tic Computing*. He is a regular reviewer of more than 20 journals including IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, IEEE TCYB, IEEE TSMCA, and *Information Sciences*.

Qizhang Luo received the B.S. and the M.S. degrees in traffic and transportation engineering, in 2015 and 2018, respectively, from Central South University, Changsha, China, where he is currently working toward the Ph.D. degree in traffic and transportation engineering.

He is also a Ph.D. visiting student with the National University of Singapore, Singapore. His current research interests include computational intelligence and scheduling, with a focus on their applications in transportation and aerospace fields.

Xiao Du received the M.S. degree in traffic and transportation engineering from Central South University, Changsha, China, in 2021.

He is currently working with Henan Transport Investment Group Company, Ltd., Zhengzhou, China. His research interests include computational intelligence and satellite scheduling.

Yingguo Chen received the B.S., M.S., and Ph.D. degrees in systems engineering from the National University of Defense Technology, Changsha, China, in 2008, 2010, and 2014, respectively.

He is currently an Assistant Professor with the College of Systems Engineering, National University of Defense Technology, Singapore. His current research interests include intelligent optimization, mission planning, and scheduling methods.

Ponnuthurai Nagaratnam Suganthan (Fellow, IEEE) received the B.A. and M.A. degrees from the University of Cambridge, Cambridge, U.K., and the Ph.D. degree in computer science from Nanyang Technological University, Singapore.

He is currently a professor with Nanyang Technological University, Singapore.

Dr. Suganthan was the recipient of the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION Outstanding Paper Award in 2012 and the Highly Cited Researcher Award by the Thomson Reuters in computer science in 2015 . He is currently an Associate Editor for the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, the IEEE TRANSACTIONS ON CYBERNETICS, *Information Sciences*, and *Pattern Recognition* and the Founding Co-Editor-in-Chief for *Swarm and Evolutionary Computation* Journal.

Xinwei Wang received the B.S. and Ph.D. degrees in aerospace engineering from Beihang University, Beijing, China, in 2013 and 2019, respectively.

He was with Queen Mary University of London, London, U.K., and is currently a Postdoc with the Delft University of Technology, Delft, Netherlands. His current interests include satellite scheduling and intelligent transport systems.