Generalized Model and Deep Reinforcement Learning-Based Evolutionary Method for Multitype Satellite Observation Scheduling

Song, Yanjie; Ou, Junwei; Pedrycz, Witold; Suganthan, Ponnuthurai Nagaratnam; Wang, Xinwei; Xing, Lining; Zhang, Yue

DOI
10.1109/TSMC.2023.3345928

Publication date
2024

Document Version
Final published version

Published in
IEEE Transactions on Systems, Man, and Cybernetics: Systems

Generalized Model and Deep Reinforcement Learning-Based Evolutionary Method for Multitype Satellite Observation Scheduling

Yanjie Song, Junwei Ou, Witold Pedrycz, *Life Fellow, IEEE*, Ponnuthurai Nagaratnam Suganthan, *Fellow, IEEE*, Xinwei Wang, *Member, IEEE*, Lining Xing, and Yue Zhang

***Abstract*—Multitype satellite observation, including optical observation satellites, synthetic aperture radar (SAR) satellites, and electromagnetic satellites, has become an important direction in integrated satellite applications due to its ability to cope with various complex situations. In the multitype satellite observation scheduling problem (MTSOSP), the constraints involved in different types of satellites make the problem challenging. This article proposes a mixed-integer programming model and a generalized profit representation method in the model to effectively cope with the situation of multiple types of satellite observations. To obtain a suitable observation plan, a deep reinforcement learning-based genetic algorithm (DRL-GA) is proposed by combining the learning method and genetic algorithm. The DRL-GA adopts a solution generation method to obtain the initial population and assist with local search. In this method, a set of statistical indicators that consider resource utilization and task arrangement performance are regarded as states. By using deep neural networks to estimate the Q value of each action, this method can determine the preferred order of task scheduling. An individual update strategy and an elite strategy are used to enhance the search performance of DRL-GA. Simulation results verify that DRL-GA can effectively solve the MTSOSP and outperforms the state-of-the-art algorithms in several aspects. This work reveals the advantages of the proposed generalized model and scheduling method, which exhibit good scalability for various types of observation satellite scheduling problems.**

***Index Terms*—Combinatorial optimization problem, deep reinforcement learning (DRL), evolutionary algorithm (EA), generalized model, multitype, satellite observation, scheduling.**

Manuscript received 14 June 2023; revised 29 September 2023; accepted 15 December 2023. Date of publication 15 January 2024; date of current version 19 March 2024. This work was supported in part by the Science and Technology Innovation Team of Shaanxi Province under Grant 2023-CX-TD-07; in part by the Special Project in Major Fields of Guangdong Universities under Grant 2021ZDZX1019; and in part by the Hunan Key Laboratory of Intelligent Decision-Making Technology for Emergency Management under Grant 2020TP1013. This article was recommended by Associate Editor N. Sun. *(Yanjie Song and Junwei Ou contributed equally to this work.) (Corresponding authors: Lining Xing; Yue Zhang.)*

Yanjie Song is with the School of Electronic Engineering, Xidian University, Xi'an 710071, China, and also with National Defense University, Beijing 100091, China (e-mail: songyj_2017@163.com).

Junwei Ou is with the Department of Computer Science and the Cyberspace Security College, Xiangtan University, Xiangtan 411105, China (e-mail: junweiou@163.com).

Witold Pedrycz is with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB T6R 2G7, Canada, also with the Systems Research Institute, Polish Academy of Sciences, 00-901 Warsaw, Poland, and also with the Faculty of Engineering and Natural Sciences, Department of Computer Engineering, Istinye University, 34010 Sariyer/Istanbul, Türkiye (e-mail: wpedrycz@ualberta.ca).

Ponnuthurai Nagaratnam Suganthan is with the KINDI Center for Computing Research, College of Engineering, Qatar University, Doha, Qatar (e-mail: p.n.suganthan@qu.edu.qa).

Xinwei Wang is with the Department of Cognitive Robotics, TU Delft, 2628 CD Delft, The Netherlands (e-mail: xinwei.wang.china@gmail.com).

Lining Xing is with the School of Electronic Engineering, Xidian University, Xi'an 710071, China (e-mail: lnxing@xidian.edu.cn).

Yue Zhang is with the School of Reliability and Systems Engineering, Beihang University, Beijing 100191, China (e-mail: zhangyue1127@buaa.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TSMC.2023.3345928.

Digital Object Identifier 10.1109/TSMC.2023.3345928

I. Introduction

WITHIN the last two decades, space technology has undergone rapid advancement, with satellite observation, communication, and navigation becoming increasingly familiar to the public [1]. Satellite observation refers to the process of acquiring images or signal characteristics of stationary or moving objects on the ground or at sea using Earth observation satellites (EOSs) payloads. These satellites have a broad range of applications in various fields, including agriculture, meteorology, oceanography, and industry [2]. Among many techniques used in satellite observation, satellite scheduling plays a crucial role. The plans obtained through satellite scheduling enable the rational and efficient use of satellite resources while meeting the needs of users as much as possible. However, the increasing demand and complexity of the environment pose significant challenges to satellite observation [3].

Optical observation satellites, synthetic aperture radar (SAR) satellites, and electromagnetic (EM) satellites are the three commonly used observation satellites [4]. Each type of satellite has its specific uses and difficulties in performing tasks in various situations [5]. Therefore, investigating how to effectively use EOSs and develop a reasonable plan for the three types of satellites is necessary. The core element of the observation plan is determining the satellites and times for executing each task that can obtain high profits. Unlike other combinatorial optimization problems, satellites have a fixed orbit and are only observable when passing over the task area. This time range over which the task can be observed is referred to as the visible time window (VTW) [6]. Therefore, the

multitype satellite observation scheduling problem (MTSOSP) involves selecting the appropriate tasks for the three types of EOSs and completing tasks within their corresponding VTWs.

The optical detection satellite scheduling problem has been extensively studied in previous works [7], [8], whereas scheduling problems for the other two types of satellites are relatively rare. There are no relevant studies on MTSOSP in the existing literature. The use of multiple types of satellites for observation missions has obvious advantages. In particular, multitype satellite observation can complement each other to overcome the influence of various factors, such as weather, clutter, and equipment capacity, and ensure the successful completion of observation tasks [9]. Nevertheless, developing an effective plan for multitype satellite observation tasks presents new challenges. Existing models that consider only one type of satellite are no longer applicable. Therefore, a new model and algorithm need to be proposed to solve the MTSOSP effectively. The research in this article will provide modeling and methodological support for multitype satellite observation scheduling.

Several evolutionary algorithms (EAs), such as genetic algorithm [10], ant colony algorithm [11], particle swarm optimization algorithm [12], and memetic algorithm [13], have been applied to solving EOS scheduling problems (EOSSP), providing effective planning for the presented scenarios in the studies. A genetic algorithm is popular among these algorithms for its simple structure and outstanding performance [14]. However, EAs, such as genetic algorithms, cannot guarantee to find high-quality solutions due to the mechanism of random search [15], [16]. In addition, the performance of the algorithm also suffers from the lack of exploitation capability. In the complex MTSOSP studied in our work, solution effectiveness and efficiency are crucial factors to consider in algorithm design. Therefore, we propose a deep reinforcement learning-based genetic algorithm (DRL-GA) for solving the MTSOSP. DRL-GA combines the respective advantages of deep reinforcement learning (DRL) methods and genetic algorithms. A Markov decision process is constructed based on task feature information to generate initial solutions for population search and neighborhood search. During the population search process, the algorithm uses an elite strategy to enhance the search speed. When the population search reaches predetermined conditions, a fast local search method is employed to find higher-quality solutions. The contributions of this study are as follows.

1) We construct a generalized mathematical model that represents the observation profits of optical, SAR, and EM satellites in a generalized form, accounting for external and internal induced influences. This profit representation method realizes the unification of the profit evaluation standard of each type of satellite. The type judgment function used in the model can effectively reduce the workload of constraint judgment. This mixed-integer planning model is an extension of the single-type EOS scheduling model. It can be applied not only to the MTSOSP but also to other EOSSP problems with some modifications.
2) A genetic algorithm based on DRL is proposed, which generates solutions using a DRL method. The initial solution is obtained heuristically using the DRL method before running the population iterative search. This DRL method can choose tasks continuously to form chromosomes based on the state. In the local search stage, a DRL-assisted heuristic task insertion method is used to quickly generate a new neighborhood structure. An elite strategy is also adopted to speed up the population search. The idea of utilizing DRL to enhance the performance of GA's global and local search can be applied to other studies on DRL-assisted EA on solving combinatorial optimization problems.
3) Our proposed DRL-GA outperforms several state-of-the-art algorithms and can effectively solve the MTSOSP at different task scales. The simulation results demonstrate that DRL-GA has superior performance in terms of profit, convergence speed, and solution time compared to other advanced algorithms.

The remainder of this article is organized as follows. In Section II, we provide a review of related work on EOSSP and methods of combining DRL with EA. In Section III, we introduce a multitype satellite observation scheduling model. In Section IV, we present the solution generation method and the DRL-GA algorithm. In Section V, we present the simulation results of the proposed algorithm and compare it with state-of-the-art algorithms. Finally, we conclude our work.

II. Related Work

A. Earth Observation Satellite Scheduling Problem

The EOS scheduling problem has received substantial attention due to its various applications in areas, such as commercial spaceflight and satellite Internet. Depending on the type of problem, it can be subdivided into the EOS scheduling problem (EOSSP), satellite measurement and control scheduling problem (SRSP), satellite data downlink scheduling problem, and satellite interplanetary routing scheduling problem. Wolfe and Sorensen [17] were among the first to use optimization models to describe the EOS scheduling problem. Later studies proved that EOSSP is NP-hard and has no polynomial time algorithm [18]. In these studies, mixed-integer models, quadratic models, and constraint satisfaction problem models have since been developed [19], [20], [21]. These models consider various factors that affect the accomplishment of the task of optical observation satellites, such as cloud cover and illumination.

While various models have been proposed for the EOS scheduling problem, various types of solution algorithms have also emerged. These algorithms can be classified as exact algorithms, heuristic algorithms, EAs, reinforcement learning algorithms, and others. Exact algorithms can obtain optimal solutions when the problem size is small, but their solution time exponentially increases as the problem size grows. Therefore, exact algorithms are typically used only for problems in specific scenarios and are not considered practical for larger-sized problems. Heuristic algorithms and

EAs, including metaheuristics, are commonly used in practical engineering applications [22]. Fatos et al. [14] showed that the genetic algorithm outperformed other search algorithms in solving the EOS scheduling problem. Xu et al. [23] used an improved genetic algorithm (IGA) to solve the large area observation scheduling problem, specifically designed for the EOSSP.

Incorporating local search with EAs is another way to improve the performance of the algorithm search and achieve better results. Due to the high complexity and large solution space of EOSSP, local search methods can help the solution algorithm to search for local optima in the local space. Chang et al. [24] proposed a memetic algorithm to solve the EOS scheduling problem, which uses neighborhood structure destruction and repair to generate new solutions during local search. Wei et al. [25] proposed a multiobjective memetic algorithm framework to obtain a high-quality agile EOS observation plan to minimize task failure and load balancing.

B. DRL Combining With EA for Solving Combinatorial Optimization Problem

Reinforcement learning methods use an agent (or multiagents) to interact with the environment and find the optimal solution by learning. Compared to traditional optimization methods, DRL has become an important approach for solving combinatorial optimization problems. The pointer networks has played a significant role in promoting DRL for solving combinatorial optimization problems [26]. However, solving combinatorial optimization problems using reinforcement learning methods alone requires a lot of effort in model training to achieve results that exceed or are close to those produced by heuristic and EAs [27]. Recent studies [28], [29] have explored the combination of reinforcement learning with EAs. In such approaches, several aspects of EAs, such as individual selection, parameter control, and population evolutionary operations, are assisted by DRL methods. Tian et al. [28] attempted to allow the agent to provide decisions for the multiobjective optimization algorithm to select the operation operator. Du et al. [29] combined DRL methods with distribution estimation algorithms to find solutions for the multiobjective hybrid shop scheduling problem. The agent's actions were various neighborhood-improving heuristics rules.

III. Mathematical Model

A. Problem Description

The MTSOSP aims to develop a task execution plan for each Optical EOS, SAR EOS, and EM EOS that includes the execution sequence and timing. The scheduling problem includes multiple tasks, with varying degrees of urgency or importance, which are reflected in the task timeliness constraint. A task is considered valid only if it meets the user's timeliness requirements, i.e., if it has started and completed within the required time frame. Only satellites that fly over the task area have the potential to execute tasks. Furthermore, the satellite must execute its task within the VTW's range where the task can be observed.

Each satellite requires energy to capture images and store them in satellite memory, which has limited capacity and cannot be replenished in real-time. Therefore, a reasonable task execution plan is required under the condition of limited resource capacity to ensure the full utilization of satellite resources. After completing an observation task, the satellite cannot immediately perform the next task but needs to go through a certain conversion time. Furthermore, the three types of satellites, Optical EOSs, SAR EOSs, and EM EOSs, carry different working policies and modes of payloads, and the working conditions to be satisfied by the payload work are not the same. Optical EOSs are affected by clouds and light [1], [30], while SAR EOSs and EM EOSs are sensitive to EM signals [9], [31], [32]. These factors are ultimately reflected in the quality of the images captured. Regardless of the type of satellite, a satellite observation is considered successful only when the image quality requirements of the user are met.

The goal of scheduling is to find a sequence of tasks that maximizes the observational profit, which is closely related to the degree of importance of the task.

B. Variables and Symbols

Parameters and decision variables are shown in Table I.

C. Model

Due to the various factors that can affect the flight and task of the satellite, some of which do not need to be considered within the scope of the model, the following assumptions are made. Based on assumptions made in [21], [33], and [34], several specific assumptions for MTSOSP are given as follows.

Assumption:

1) Observation tasks are preprocessed to cover the entire task area in a single observation [33].
2) Satellite energy and memory can be restored to their initial state at the beginning of each orbital flight [34].
3) Each task can be executed at most once and does not need to be repeated [21].
4) A task can be successfully scheduled and then executed without being affected by other factors that could cause failure [21].
5) The task and its specific execution requirements are defined before scheduling starts, and no new tasks or temporary cancellations occur [21].
6) Each satellite can execute an observation task at any moment [34].

The diverse measure used to assess the observational profits of different satellite types pose challenges in comparing the advantages and disadvantages of executing the same task with varying satellites. To achieve the evaluation of different satellite types profits on the same measure, we propose a generalized method for observational profits representation. This method serves as a premise for proposing the MTSOSP model. This method involves categorizing factors that impact different types of satellites. These factors are relatively independent

TABLE I
VARIABLES AND SYMBOLS

Symbol	Description
Sat	Set of observation satellite resources
T	Set of observation tasks
TW	Set of observation time windows
O_i	Set of orbits belonging to satellite i
$type_i$	Type of satellite i
Mem_i	Memory capacity of satellite i
Eng_i	Power limit of satellite i
$mode_i$	Observation modes supported by satellite i
res_i	Resolution supported by satellite i
$band_i$	Bandwidth settings supported by satellite i
fre_i	Band settings supported by satellite i
pol_i	Polarization mode supported by satellite i
ϑ_i	Maximum observation angle supported by satellite i
LS_i	Load on/off time of satellite i
clo_j	Cloud information of task j
env_j	Electromagnetic environment situation of task j
$a_j^{\max}$	Maximum allowable observation angle of task j
d_j	Observation duration of task j
mem_j	Memory consumption of task j
eng_j	Power consumption of task j
r_j^t	Satellite type requirement of task j
r_j^m	Observation mode setting requirement of task j
r_j^b	Bandwidth setting requirement of task j
r_j^b	Polarization method setting requirement of task j
r_j^r	Resolution setting requirement of task j
r_j^f	Frequency setting requirement of task j
$Ctype\,(\cdot)$	Task type judgment function
$Cband\,(\cdot)$	Task bandwidth judgment function
$Cres\,(\cdot)$	Task Resolution judgment function
$Cpol\,(\cdot)$	Task polarization judgment function
$Cfre\,(\cdot)$	Frequency setting judgment function
$Cclo\,(\cdot)$	Cloudiness judgment function
$Cenv\,(\cdot)$	Electromagnetic environment judgment function
a_{ij}^t	Observation angle of the task j located at the satellite i at moment t
$revt_{ijko}$	Earliest visible time of satellite i in orbit o for task j at the kth time window
$rlvt_{ijko}$	Latest visible time of satellite i in orbit o for the kth time window of task j
$tr_{ijj'}$	transition time of satellite i between task j and task j'
$rest_j$	Task j earliest allowed start time
$rlet_j$	The latest allowed end time for task $task_j$
M	A large integer
x_{ijko}	Whether the ith satellite performs the jth task at the kth time window of its oth orbit, if so, $x_{ijko}=1$; otherwise, $x_{ijko}=0$
st_{ijo}	Start time of the jth task of the ith satellite in its oth orbit

and are computed using a standardized formula to derive the actual observation profits under a consistent measure. Specifically, factors affecting the observation profit of satellites are divided into two categories: external environmental factors (mainly including factors related to cloud cover [35], EM environment [32], etc.) and internal factors (mainly, including the factors related to observation satellites [36], tasks [7], etc.). Both external environmental factors and internal factors will have an impact on the observation profit. The original observation profit of an observation task, task_j, is denoted by opro_j, which is also the maximum observation profit obtained in the best case. Then, we can calculate the actual observation profit, $apro_{ijk}$, for each type of satellite, sat_i, by combining the maximum observation profit with the external environmental factors and internal factors. The equation for calculating the actual observation profit is shown

$$p_{ijk} = \text{opro}_j \times OI_{ijk} \times II_{ijk} \tag{1}$$

where opro_j denotes maximum observation profit for task task_j, OI_{ijk} denotes the influence of external environmental factors, II_{ijk} denotes the influence of internal factors, OI_{ijk} and II_{ijk} and both in the range of [0, 1].

The actual observation profit depends on the satellite resources and the time interval in which the task is executed. The effect of external environmental factors can be expressed as follows:

$$OI_{ijk} = \prod_{l=1}^{N_{OI}} \left[1 - oi_l\left(\text{sat}_i, \text{task}_j, tw_k\right)\right] \tag{2}$$

where $oi_l(\cdot)$ denotes the function affected by external environmental factors when the satellite sat_i performs the task task_j, and N_{OI} denotes the number of satellites of each type affected by external environmental factors. A smaller value of $oi_l(\text{sat}_i, \text{task}_j, tw_k)$ indicates a smaller influence of this factor. When the task is not affected by external environmental factors, the value of the corresponding function term is 0.

Similarly, the effect of internal factors can be expressed as follows:

$$II_{ijk} = \prod_{l=1}^{N_{II}} \left[1 - ii_l\left(\text{sat}_i, \text{task}_j, tw_k\right)\right] \tag{3}$$

where $ii_l(\cdot)$ denotes the lth internal factor influence function expression, and N_{II} denotes the number of satellites of each type influenced by internal factors. A smaller value of $ii_l(\text{sat}_i, \text{task}_j, tw_k)$ indicates a smaller influence by the factor, and the opposite is true for a larger influence. When the observation task is not affected by the lth internal factor, the value of the corresponding function term is 0.

By using the observation profit generalized description method, the task profits belonging to different types of satellite execution tasks can be transformed into the same determination criteria. After completing the representation of the observation task profit in a generalized form, the objective function can be consistently constructed. The goal of our optimization is to find a task sequence that maximizes the observation profit, i.e., maximizes the observation profit. The objective function for the MTSOSP can be expressed as follows:

Objective Function:

$$\max \sum_{i\in Sat} \sum_{j\in T} \sum_{k\in TW} \sum_{o\in O_i} p_{ijk} \cdot x_{ijko} \tag{4}$$

where p_{ijk} denotes the profit and x_{ijko} denotes whether the satellite i executes the task j in the time window k of the orbit o.

There are numerous constraints for each type of satellite. Determining whether or not these constraints are satisfied can result in task scheduling that takes a lot of time. Therefore, it is necessary to propose a satellite type-task judgment function to simplify the workload of constraint judgment. The satellite type-task type matching judgment function serves as the basis for determining specific task requirements. When the satellite

type does not match the type of task requirements, there is no need to make further judgments on whether the satellite satisfies the specific constraints. This can reduce the workload of judging whether the constraints are satisfied to a certain extent. Only when the satellite type matches the type of task requirements, further judgments of more specific constraints are made. The specific representation of the satellite type-task type matching judgment function can be expressed as

$$\mathrm{Ctype}\left(\mathrm{type}_i, r_j^t\right) = \begin{cases} 1, & \text{if both types are consistent} \\ 0, & \text{else.} \end{cases} \tag{5}$$

For ease of representation, the results of the judgment function can be abbreviated to the following form:

$$\Delta \leftarrow \mathrm{Ctype}\left(\mathrm{type}_i, r_j^t\right). \tag{6}$$

The type judgment function is swiftly screen potential to execute a task, as many tasks have corresponding relationships with specific satellite types. In cases where the type matching relationship is not satisfied, the model will refrain from attempting to task arrangement. This approach effectively reduces the workload involved in task scheduling. Furthermore, the observation mode, bandwidth setting, resolution setting, polarization mode, frequency band setting, cloud amount, and EM environment influence can be checked by the corresponding judgment function to determine if the task execution requirements are met. If the requirements are satisfied, the function return value is 1; otherwise, it is 0.

Based on the above judgment functions, it is easy to determine whether the satellite can meet the requirements of task execution. Next, we will provide the satellite capability constraints that need to be satisfied for collaborative scheduling in general. These constraints form a further extension of constraints presented in the single type EOS observation scheduling problem [37], [38]. The constraints in the model are independent, which can be flexibly utilized according to the specific problem. When a specific problem is to be solved, some of the constraints can be added, modified, or discarded flexibly as needed.

Constraints:

$$x_{ijko} \le \Delta \quad \forall i \in \mathrm{Sat}, j \in T, k \in TW, o \in O_i \tag{7}$$

$$\Delta \cdot x_{ijko} \le Cm\left(\mathrm{mode}_i, r_j^m, \Delta\right) \quad \forall i \in \mathrm{Sat}, j \in T, k \in TW, o \in O_i \tag{8}$$

$$\Delta \cdot x_{ijko} \le Cb\left(\mathrm{band}_i, r_j^b, \Delta\right) \quad \forall i \in \mathrm{Sat}, j \in T, k \in TW, o \in O_i \tag{9}$$

$$\Delta \cdot x_{ijko} \le Cr\left(\mathrm{res}_i, r_j^r, \Delta\right) \quad \forall i \in \mathrm{Sat}, j \in T, k \in TW, o \in O_i \tag{10}$$

$$\Delta \cdot x_{ijko} \le Cp\left(\mathrm{pol}_i, r_j^p, \Delta\right) \quad \forall i \in \mathrm{Sat}, j \in T, k \in TW, o \in O_i \tag{11}$$

$$\Delta \cdot x_{ijko} \le Cf\left(\mathrm{fre}_i, r_j^f, \Delta\right) \quad \forall i \in \mathrm{Sat}, j \in T, k \in TW, o \in O_i \tag{12}$$

$$\Delta \cdot x_{ijko} \le Cc\left(\mathrm{clo}_j, \Delta\right) \quad \forall i \in \mathrm{Sat}, j \in T, k \in TW, o \in O_i \tag{13}$$

$$\Delta \cdot x_{ijko} \le Ce\left(\mathrm{env}_j, \Delta\right) \quad \forall i \in \mathrm{Sat}, j \in T, k \in TW, o \in O_i \tag{14}$$

$$\sum_{j \in T \setminus \{j'\}} \sum_{k \in TW} \mathrm{mem}_j \cdot x_{ijok} + \mathrm{mem}_{j'} \cdot x_{ij'ok'} \le \mathrm{Mem}_i \quad \forall i \in \mathrm{Sat}, j \in T, o \in O_i, k, k' \in TW \tag{15}$$

$$\sum_{j \in T \setminus \{j'\}} \sum_{k \in TW} \mathrm{eng}_j \cdot x_{ijok} + \mathrm{eng}_{j'} \cdot x_{ij'ok'} \le \mathrm{Eng}_i \quad \forall i \in \mathrm{Sat}, j \in T, o \in O_i, k, k' \in TW \tag{16}$$

$$st_{ijo} \le \mathrm{revt}_{ijko} \cdot x_{ijko} \quad \forall i \in \mathrm{Sat}, j \in T, k \in TW, o \in O_i \tag{17}$$

$$\left(st_{ijo} + d_j\right) \cdot x_{ijko} \le rlvt_{ijko} \quad \forall i \in \mathrm{Sat}, j \in T, k \in TW, o \in O_i \tag{18}$$

$$a_{ij}^t \cdot x_{ijko} \le \min\left\{\vartheta_i, a_j^{\max}\right\} \quad \forall i \in \mathrm{Sat}, j \in T \quad k \in TW, o \in O_i, t \in \left[st_{ijo}, st_{ijo} + d_j\right] \tag{19}$$

$$\left(st_{ijo} + d_j\right) \cdot x_{ijko} + tr_{ijj'} \le st_{ij'o} + I \cdot \left(1 - x_{ij'k'o}\right) \quad \forall j \ne j', i \in \mathrm{Sat}, j, j' \in T, o \in O_i, k, k' \in TW \tag{20}$$

$$\sum_{i \in Sat} \sum_{k \in TW} \sum_{o \in O_i} x_{ijko} \le 1 \quad \forall i \in \mathrm{Sat}, j \in T, k \in TW, o \in O_i \tag{21}$$

$$x_{ijko} \in \{0, 1\} \quad \forall i \in \mathrm{Sat}, j \in T, k \in TW, o \in O_i \tag{22}$$

$$st_{ijo} \in Z^* \quad \forall i \in \mathrm{Sat}, j \in T, k \in TW, o \in O_i. \tag{23}$$

Equation (7) indicates that the satellite needs to be consistent with the type required by the task. Equation (8) indicates that the observation mode used by the satellite needs to be the same as the observation mode required by the task. Equation (9) indicates that the bandwidth used needs to be the same as the bandwidth setting required by the task. Equation (10) indicates that the resolution used needs to be guaranteed to be the same as the resolution setting required by the task. Equation (11) indicates that the polarization method used needs to be the same as the one required by the task. Equation (12) indicates that the frequency used needs to be the same as the frequency setting required by the task. Equation (13) indicates that the satellite executes tasks that cannot be affected by cloud cover. Equation (14) indicates that the satellite executes tasks that cannot be affected by the EM environment. Equation (15) indicates that the satellite cannot exceed the upper limit of the satellite's memory capacity for each orbit in which it flies to complete its task. Equation (16) indicates that the satellite cannot exceed the upper limit of satellite power in each orbit to complete its task. Equations (17) and (18) indicate that each task needs to be executed within a time window that can be detected. Equation (19) indicates that the observation angle to the task needs to be less than the maximum allowable angle. Equation (20) indicates that the interval time requirement needs to be satisfied for the transition between two tasks. Equation (21) indicates that each task can be executed at most once. Equations (22) and (23) indicate the value range of decision variables.

IV. Deep Reinforcement Learning-Based Genetic Algorithm

A. Framework

To address the MTSOSP, we propose the DRL-GA, which uses DRL methods to generate solutions for population search and local search. The overall framework of the algorithm is shown in Fig. 1. The genetic algorithm that successfully solves multiple satellite scheduling problems is very popular for its simple structure and excellent exploration capability. These metrics drive us to propose a DRL-GA for the observation scheduling problem with multiple types of satellites. DRL-GA involves four innovations.

1) A DRL-based initialization heuristic is proposed. The DRL method constitutes individuals in the order of

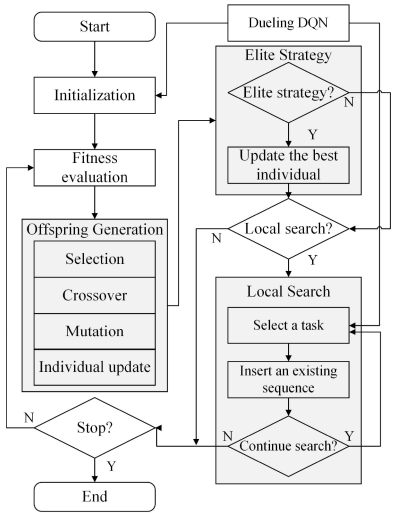

Fig. 1. Framework of DRL-GA.

Algorithm 1: Solution Generation Method Based on DRL

Input: task set T, population size N_p, Dueling DQN, ε_g
Output: population P_0

```
1  for i = 1 to N_p do
2      if rand() ≥ ε_g then
3          while termination criterion is not met do
4              a_t ←Choose action by Dueling DQN(S_t, A);
5              task ←Select the task according to Action
               Selection Method (T, a_t);
6              Omit task from T;
7              indi_i ←Add task into individual i;
8              R_t ←Calculate Reward;
9              S_{t+1} ←Update State;
10             Relay Buffer←Record state transition;
11             t ← t + 1;
12     else
13         indi_i ←Random generate an individual;
14     P_0 ←Add indi_i into P_0;
```

preference for selecting a task based on a state consisting of multiple statistical values.

2) An individual update mechanism is used to select individuals to generate offspring.
3) An elite individual retention strategy is embedded in the algorithm framework. The best individuals within the population are directly retained in the offspring during the initial search stage to improve the search efficiency.
4) A fast local search method is proposed to enhance the exploitation capability of the algorithm. This local search method uses a DRL-assisted method to select tasks quickly and construct a new neighborhood structure.

The DRL-GA proposes a novel algorithmic design concept for DRL, aiming to enhance the performance of both GA population search and local search simultaneously.

B. Solution Generation Method Based on DRL

In our proposed algorithm, we utilize the DRL method to generate the initial solution for the population search of GA and obtain the task sequence used by the local search to construct new neighborhood solutions. In previous studies, various methods, such as random search, specific heuristic rules, and machine learning, have been used to construct solutions [39]. However, these methods can be too random or only effective for certain scenarios or problems, leading to weak generalization performance. In contrast, DRL is a method with strong generalization ability and has been successfully applied to a variety of combinatorial optimization problems. By using DRL methods, we can effectively generate high-quality initial solutions and improve the algorithm's ability to find optimal solutions. In the MTSOSP, the choice of which task to plan is only related to the current state, satisfying the requirement of no posteriority in the construction of the Markov decision process. A Markov decision process can be composed of four components $\langle S, A, R, V\rangle$, where S denotes the state, A stands for the action, R is the reward, and V denotes the value function. The state represents the situation of the agent itself at time step t and is generally represented by a feature matrix. The solution generation algorithm is presented in Algorithm 1, which generates a sequence of tasks based on the current state using the DRL method.

As shown in Algorithm 1, DRL or randomized method can be used to generating individual chromosomes (Line 2–13). When an individual generates a chromosome sequence using DRL, the neural network obtains the Q-value for each action strategy based on the input state (Line 4). The definitions of states and actions are presented in Section IV-B1 and IV-B2, respectively. Subsequently, an action selection method as shown in Algorithm 2 will be used to select a strategy (Line 5). This strategy will sort the optional tasks and the task in the top position will be used to generate the chromosome sequence.

1) State: The state of an agent serves as the foundation for calculating the Q value using deep neural networks and selecting an action. The state space S is a sequence of states S_t, defined as

$$S = \{S_0, S_1, \ldots, S_t, \ldots\} \tag{24}$$

where S_t denotes the state of the agent at time step t. In our approach, each state S_t is a composition of statistical indicators with attribute values related to the scheduling results. These indicators effectively describe the agent's performance during the construction of the solution to the MTSOSP. The attributes constituting the state S_t are related as follows:

$$S_t = \{\mathrm{RAT}_t, \mathrm{RSTD}_t, \mathrm{RAP}_t, \mathrm{RAUP}_t\} \tag{25}$$

where RAT_t denotes the total time available for the remaining time window, RSTD_t denotes the standard deviation of the average of the remaining tasks from the average of the profit of all tasks, RAP_t denotes the average profit of the remaining tasks, and RAUP_t denotes the average profit per unit time of the remaining tasks. We calculate the total hours available for the remaining time window, the standard deviation of the average of the remaining tasks and the average of the profits of all tasks, the average profit value of the remaining tasks,

and the average profit value per unit time of the remaining tasks using

$$\mathrm{RAT}_t = \sum_{i\in Sat}\sum_{j\in T}\sum_{k\in TW}\sum_{o\in O_i}\left(rlvt_{ijko} - \mathrm{revt}_{ijko}\right) - \sum_{i\in Sat}\sum_{j\in ST_t}\sum_{k\in TW}\sum_{o\in O_i} d_j \cdot x_{ijko} \tag{26}$$

$$\mathrm{RSTD}_t = \frac{1}{|RT_t|}\sqrt{\sum_{j\in RT_t}\left(\mathrm{opro}_j - \mathrm{opro}_{\mathrm{ave}}\right)^2} \tag{27}$$

$$\mathrm{RAP}_t = \frac{1}{|RT_t|}\sum_{j\in RT_t}\mathrm{opro}_j \tag{28}$$

$$\mathrm{RAUP}_t = \sum_{j\in RT_t}\mathrm{opro}_j \Big/ \sum_{j\in RT_t} d_j \tag{29}$$

$$\mathrm{opro}_{\mathrm{ave}} = \frac{1}{|T|}\sum_{j\in T}\mathrm{opro}_j \tag{30}$$

where RT_t denotes the set of remaining tasks at the time step t.

2) Action: Action selection determines the preferred order of task scheduling in the Markov model. The exact order slightly differs between generating the initial solution for population search and the initial solution for local search. One action selection identifies a task to be planned and places it after the previous action selection's task. If applied to local search, the selected task also needs to be rejoined to the task sequence according to a certain strategy to obtain a new neighborhood structure.

Our action selection strategy is based on heuristic rules, and the corresponding task is selected from the optional task sequence according to these rules. We use four action strategies, which are as follows.

Profit-First Strategy: Select the observation task with the highest profit in the remaining set of optional tasks RT.

Unit Time Profit-First Strategy: Select the observation task with the highest unit time profit value in the remaining set of optional tasks RT, and the task unit time profit is calculated as $up_j = \mathrm{opro}_j/d_j$, where opro_j denotes the task profit and d_j denotes the task requirement duration.

Time Urgency-First Strategy: Select the observation task with the highest time urgency requirement among the remaining set of optional tasks RT.

Duration-First Strategy: Select the observation task with the shortest duration in the remaining set of optional tasks RT.

A balance between exploration and exploitation is necessary for action selection. Exploration focuses on the DRL method's global search capability, while exploitation focuses on its local search capability. Algorithm 2 shows the pseudocode for action selection. In this algorithm, ε_c represents a control parameter to determine whether the action selection is taken in a greedy way (Line 3) or in a random way (Line 5).

3) Reward: The reward evaluates the agent's performance in taking action A_t in state S_t. In the MTSOSP, the agent calculates the reward value obtained for each action choice based on fitness improvement. Specifically, the rewards at time step t are obtained by subtracting the fitness values at time step

Algorithm 2: Action Selection Method

Input: state S_t, Dueling DQN, action set A, ε_c
Output: action a_t
1 *rand* ←Generate a random number between 0 and 1; **if** *rand* $\geq \varepsilon_c$ **then**
2 $\quad A_t$ ←Choose the action with the largest Q value from A;
3 **else**
4 $\quad A_t$ ← Random choose an action from A;

t and time step $t-1$. The equation for calculating rewards is

$$R_t = \mathrm{fit}_t - \mathrm{fit}_{t-1} \tag{31}$$

where fit_t denotes the value of the fitness function at time step t and fit_t denotes the value of the fitness function at time step $t-1$. The adaptation function value is calculated by the DTTWSA algorithm in [40] according to (4).

4) State Transition: When the agent takes an action A_t at time step t according to state S_t, it will move to the next state S_{t+1}. The subsequent action selection will be based on the state S_{t+1}. Meanwhile, the quaternions of $\langle S_t, A_t, R_t, S_{t+1}\rangle$ are recorded in the buffer.

5) Dueling DQN-Based Value Function: We use Dueling deep Q network (DQN) to represent the value function. Dueling DQN is a type of DQN that improves the data flow on DQN by separating the state values from the rewards generated by the actions. Compared with DQN networks, Dueling DQN networks split the unidirectional data stream into two, which are used to calculate the State Value Function and Advantage Function, respectively. The calculation process is done by the fully connected network. Finally, the estimated value of Q for each action is obtained through a special aggregating layer. The Q-value function in the Dueling DQN network is represented by

$$Q^{\pi}(s,a) = V^{\pi}(s) + A^{\pi}(s,a) \tag{32}$$

where $V^{\pi}(s)$ denotes the state value and $A^{\pi}(s,a)$ denotes the action advantage.

We use the improved form of the Dueling DQN network proposed by Wang et al. [41]. This is a smoother calculation that allows the dominance function to be guided by the correct trend without pursuing the optimal case too much. The Q-value function in the improved form is represented as shown in

$$Q(s,a;\theta,\theta_v,\theta_a) = V(s;\theta,\theta_v) + \left(A(s,a;\theta,\theta_a) - \frac{1}{|\mathcal{A}|}\sum_{a'}A\left(s,a';\theta,\theta_a\right)\right) \tag{33}$$

where θ_v and θ_a are the network parameters of the two fully connected layers.

6) Dueling DQN Model Training: The neural network model can effectively find the best solution after effective learning. Dueling DQN network training adopts the same form as DQN, which is a modified form of the DQN algorithm. Dueling DQN network training has two main features. One feature is the Replay Buffer mechanism, which belongs to the Dueling DQN network model training. Dueling DQN records the agent's state transitions (S_t, A_t, R_t, S_{t+1}) and stores them in the Experience Replay Pool. Another feature is that Dueling

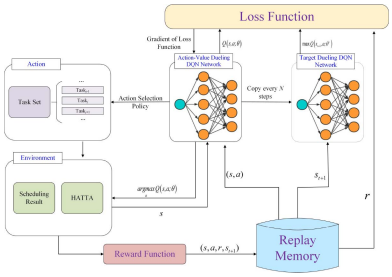

Fig. 2. Dueling DQN network training schematic.

Algorithm 3: Dueling DQN Training Method

Input: Relay Buffer, Dueling DQN, time step t, step interval SI, Batch size B_s, γ
Output: Updated action-value function Q-Network θ
1 **if** $mod(t, SI) == 0$ **then**
2 $\quad \hat{Q} \leftarrow$Copy the parameters of the Q network;
3 Sample a batch of state transition data from the Relay Buffer randomly;
4 $PV \leftarrow \hat{Q}\left(\phi_i', a'; \hat{\theta}\right)$;
5 Calculate Y in Batch by Eq. (34)-(35);
6 Calculate the loss function by $(Y_i - Q(\phi_i, A_i; \theta))^2$ and use the optimizer for gradient descent to optimize θ;

DQN learning uses a target-value function network in addition to the action-value function network. The network parameters are obtained by copying the action-value function network parameters at a fixed number of steps SI.

In Dueling DQN training, a mini-batch of state transition data (B_s) is randomly sampled from the relay buffer. The target value Y is computed by replicating the target-value function network using the mini-batch. Then, the loss function between Y and the result obtained by the action-value function network is computed, and θ is optimized by gradient descent using an optimizer. The schematic diagram of the Dueling DQN network training is shown in Fig. 2. The equations for calculating Y values during network training are shown in (34) and (35).

If S_{t+1} is not a termination state, Y is calculated as

$$Y = R_t + \gamma \max(PV). \tag{34}$$

If S_{t+1} is the termination state, Y is calculated as

$$Y = R_t \tag{35}$$

where Y denotes the predicted value of the target-value network.

The training pseudocode for Dueling DQN is shown in Algorithm 3.

Each training is done using a small batch sampling of state transition data completed from the relay buffer (Line 4). The Q-value is predicted based on the target network state value (Line 5). Then, it is determined whether it is a termination state and the corresponding Y-value calculation method is used (Line 6). If the termination condition is not reached, the Y-value is calculated according to (34). If the termination condition is reached, the reward value R_t is directly assigned to Y. After the Y-value is calculated, the parameter θ is optimized by gradient descent using the optimizer (Line 7).

C. Population Evolution Strategy

1) Crossover: Crossover and mutation form the core parts of DRL-GA. We use a double-point crossover to generate offspring. Specifically, we select two positions from the chromosome of an individual as the starting points of the crossover fragments.

2) Mutation: Mutation generates a new individual by selecting two genes at different positions in an individual and swapping their positions. Compared to crossover, the magnitude of variation can be smaller.

3) Individual Update: We propose an individual update strategy to judge whether to update the previous individuals in the population by evaluating the fitness value. If the fitness function value has improved, the individuals will be updated; otherwise, no updates will occur. However, such a greedy strategy may produce detrimental effects on the search. Therefore, we adopt the ε_u-greedy idea and introduce a threshold ε_u. When an individual update is used, a random number is generated, and when the random value is smaller than ε_u, a new individual is added to the population regardless of whether the fitness function value is improved or not.

4) Elite Strategy: To further accelerate the convergence of the algorithm, we design an elite strategy in DRL-GA. The elite strategy allows the individuals with the best performance in the search process to be effectively retained and continue to search for higher-quality solutions through evolutionary operations in the next generation of populations. However, although the elite strategy can improve the search efficiency of the population to a certain extent, it is not very meaningful to keep repeating such an operation when the population search is bottlenecked. Therefore, we introduce a mechanism to judge whether to adopt the elite strategy or not. A threshold $Thre_2$ is used to determine whether to continue using the elite strategy. A new variable $count_2$ records whether the contemporary population search has found a higher quality solution. If not, then the value of $count_2$ is increased by one. When $count_2$ equals the threshold $Thre_2$, the elite strategy is no longer used.

5) Local Search: Local search can improve the search effectiveness of algorithms, but it often requires significant computational costs. In DRL-GA, we design a low-computational cost local search algorithm by combining the solutions generated by DRL. We use a DRL method and a random task insertion method together to achieve a simple and efficient neighborhood structure improvement. At each time step t during the search process, the DRL method selects an appropriate task based on the state S_t. Subsequently, the chosen task will be inserted randomly. In this way, new solutions will be continuously constructed.

Slightly different from the task selection for generating the initial solution, the local search action selection chooses the tasks to be reinserted into the sequence, and the new neighborhood structure needs to be obtained by task insertion.

Fig. 3. Example for encoding.

Algorithm 4: DRL-GA

Input: population size N_p, α, β, task set T, time window set TW, Dueling DQN, crossover operator C_o, mutation operator M_o, step interval SI_c, crossover length L, max generation Gen, $Thre_1$, $Thre_2$, ε_u
Output: Solution S

1 Set $count_1 = 0$, $count_2 = 0$;
2 Generate an initial population by Dueling DQN;
3 **for** $gen = 1$ *to* Gen **do**
4 **for** i=1 *to* N_p **do**
5 **if** $rand() \leq \alpha$ **then**
6 $indi_i' \leftarrow$Crossover($indi_i, C_o, \alpha, L$);
7 **if** $rand() \leq \beta$ **then**
8 $indi_i' \leftarrow$Mutation($indi_i, C_m, \beta$);
9 $local_best, local_best_indi \leftarrow$Calculate population fitness value;
10 **if** $local\_best > gobal\_best$ **then**
11 $gobal\_best\_ind \leftarrow loc\_best\_indi$;
12 $gobal\_best \leftarrow loc\_best$;
13 $count_1 \leftarrow count_1 + 1$;
14 **else**
15 $count_2 \leftarrow count_2 + 1$;
16 **if** $count_1 == Thre_1$ **then**
17 Local search using DRL and insertion rules;
18 Reset $count_1 \leftarrow 0$;
19 **if** $count_2 < Thre_2$ **then**
20 Use elite strategy;
21 Use individual update strategy when $rand() < \varepsilon_u$;

TABLE II
SATELLITE ORBIT PARAMETERS

Parameter	Value
Semimajor axis (km)	6500
Inclination (°)	0.00015
Right ascension of the ascending node (°)	98.15
Eccentricity	0
Argument of perigee (°)	15.75
Mean anomaly (°)	164.25

D. DRL-GA

This section introduces the algorithm flow of DRL-GA, which uses the integer encoding method. Each integer represents the number of the task in the task set. Fig. 3 gives an example of an integer code with six tasks to be scheduled. Each gene position is a task, "6" denotes the sixth task in the task sequence, "4" denotes the fourth task in the task sequence, and so on. The fitness value is obtained according to the objective function using (4), and individual selection is done by roulette selection. The following section focuses on the crossover, mutation, individual update, and elite strategies. The pseudo-code of the genetic algorithm based on DRL is shown in Algorithm 4.

In DRL-GA, the DRL method generates initial solutions that are used in the population search and local search (Lines 3 and 26). After the population has evolved, a judgment is made on whether to replace the optimal individual based on the improvement of the fitness function value (Lines 10–15). After one generation of population search is completed, the algorithm decides whether to use the population perturbation strategy (Line 32). Additionally, DRL-GA enters the local search stage after a certain number of population searches (Lines 25–28). To generate the satellite observation plan from the DRL-GA population, a task arrangement algorithm in [40] is used.

E. Complexity of DRL-GA

The time complexity of DRL-GA in the training mode is $O(Epoch * Gen * |T| * |TW| + |T| * |TW|) = O(Epoch * Gen|T| * |TW|)$. The time complexity of DRL-GA in test mode is $O(|T| * |TW| + Gen * N_p * (|T| * |TW| + |T| * |TW|)) = O(Gen * N_p * |T| * |TW|)$. The overall space complexity of the DRL-GA is $O(N)$.

V. SIMULATION STUDIES

The simulations reported in this article were conducted on a desktop computer with a Core I7-7700 3.6 GHz CPU, 16 GB of memory (DDR4 2400 MHz), and a Windows 11 operating system, using Python 3.9.7. All algorithms were run under the same system configuration.

A. Simulation Settings

Instance Setups: Since there is no public benchmark available for the MTSOSP, we generated a certain number of instances with random tasks from around the world. To distinguish between the different instances, we used an "A-B" format, where A indicates the number of tasks in the instance and B indicates the instance number. The information of instances and the orbital parameters of one of the satellites are shown in Tables II and III.

Comparison Algorithms: In our simulations, we compared our proposed DRL-GA with a series of state-of-the-art algorithms commonly used in EOSSP and other combinatorial optimization problems. The comparison algorithms include the IGA [33], knowledge-based genetic algorithm (KBGA) [42], dual-population artificial bee colony algorithm (DPABC) [43], tabu-based adaptive large neighborhood search algorithm (ALNS-TI) [34], and neighborhood search algorithm (NS) [44]. The algorithm parameters are set as shown in Table IV.

Evaluation Metrics: To ensure the fairness of the simulation, we ran each algorithm 30 times. We evaluated the overall performance of the algorithms using the best value (denoted as Best) and the average value (denoted as Ave) of the results. We also conducted the Wilcoxon rank sum test to determine whether there was a significant difference between the search results of different algorithms, at a significance level of $p = 0.05$.

B. Results

First, the correctness of the generalized model was verified by comparing it with a model from [34]. Table V shows the scheduling performance of using the basic genetic algorithm to solve the generalized model and the CPLEX solver to solve the comparison model for 100 and 200 task scale scenarios. The results indicate that the generalized model is correct and can obtain a profit not inferior to the CPLEX solver.

TABLE III
INFORMATION OF INSTANCES

Scheduling horizon	Number of tasks	Task profit	Task duration	Number of satellites	Transition time
24h	[100,1000]	$U(5, 20)$	$N(10, 100)$	[3,15]	10s
Number of payloads per satellite	Number of task types	Number of bandwidth types	Number of polarization types	Number of resolution types	Number of frequency types
1	3	4	2	4	3

TABLE IV
PARAMETERS OF ALGORITHMS

Algorithm	Parameters
DRL-GA	$Gen = 500$, $\alpha = 0.95$, $\beta = 0.1$, $\varepsilon_u = 0.1$, $N_p = 10$, $Thre_1 = 20$, $Thre_2 = 200$, $\varepsilon_c = 0.1$ is set to 0.1, the number of hidden layers 4, the activation function ReLU, learning rate 0.00001, the discount factor 0.85.
IGA [33]	$Gen = 500$, $\alpha = 0.9$, $\beta = 0.05$, $N_p = 10$, perturbation threshold 10
KBGA [42]	$Gen = 500$, $\alpha = 0.9$, $\beta = 0.05$, $N_p = 10$, crossover length 2,3,4, initial score 50, score 30,20,10, adaptive threshold 50
DPABC [43]	$Gen = 500$, $N_p = 10$, the minimum population size of search population 3, scout bee size 1, control parameter limit 20, the degree of fitness acceptance 0.9
ALNS-TI [34]	maximum fitness evaluation times 5000, maximum iteration of no improvement 1000, percent of tasks to remove 10%, weight update parameter 0.5, coefficient of annealing 0.9975, score increment 110, 20, 30
NS [44]	maximum fitness evaluation times 5000, length 2

TABLE V
SCHEDULING PERFORMANCE OF BASIC GA AND CPLEX SOLVER FOR SMALL-SCALE SCENARIOS

Scenario	Basic GA	CPLEX Solver
100-1	**609**	**609**
100-2	**589**	**589**
100-3	**619**	**619**
200-1	**1058**	1056
200-2	**1092**	1087
200-3	**1061**	1055

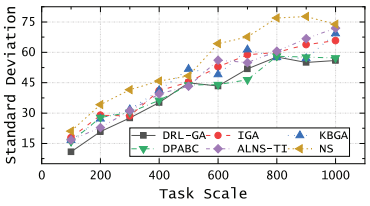

Fig. 4. Comparison on the standard deviation of algorithms.

Next, Table VI shows the scheduling performance of algorithms evaluated in different scale instances. The results demonstrate that the DRL-GA algorithm achieved the best search performance in most of the scenarios, outperforming other algorithms in terms of the average profit value and standard deviation, as well as finding the maximum profit value of the scenario. The performance gap between algorithms tended to increase as the task scale increased, reflecting the ability of DRL-GA to effectively balance exploration and exploitation. Other compared algorithms may focus too much on one aspect of exploration or exploitation, resulting in less than optimal values of the fitness function for the obtained solution. The Wilcoxon rank sum test results demonstrated a significant difference between the proposed algorithm and the other algorithms at the $p = 0.05$ level. Fig. 4 shows the average standard deviation of the algorithms at different scales. It can be seen that the standard deviation of all algorithms shows an increasing trend as the task size increases. The DRL-GA has a more stable performance in solving MTSOSP compared with IGA, KBGA, DPABC, ALNS-TI, and NS.

Then, we also analyzed the convergence performance of the algorithms, as shown in Fig. 5. The DRL-GA demonstrated a fast convergence rate in scenarios with task scales of 600, 800, and 1000, and the algorithm could quickly find solutions with high quality and continuously find solutions with higher fitness function values by flexibly using multiple search strategies.

Finally, the CPU time required for the algorithm runs are compared, and the results are shown in Table VII. Since DRL-GA uses the DRL method which consumes more system computational resources compared to other simple strategies, the algorithm time is slightly longer than the time of several compared genetic algorithms. However, compared to other search algorithms, the search time is shorter in most scenarios, with the shortest time for individual scenarios being for NS.

Task completion is also an important factor for evaluating the algorithm's ability to solve the MTSOSP. Fig. 6 shows the overall task completion rate, which demonstrates a decreasing trend as the task size increases. This is because the satellite capacity is limited and cannot complete all the tasks. DRL-GA exhibited obvious advantages over other algorithms in terms of task completion.

To verify the effectiveness of the strategies in DRL-GA, we also conducted comparative simulations on the scheduling effects by using DRL-GA without an elite strategy (denoted as DRL-GA/W1) and DRL-GA without neighborhood search (denoted as DRL-GA/W2). The results are shown in Fig. 7, which indicate that DRL-GA with elite strategy and neighborhood search can obtain higher scheduling profits compared to DRL-GA with some strategies removed. Among the strategies used by the algorithm, the enhanced search performance effect exerted by the elite strategy is more obvious, which is highly related to the complex structure of the MTSOSP solution space.

C. Discussion

The simulation results demonstrate that DRL-GA has better performance and convergence speed compared to the comparative state-of-the-art algorithm, which indicates that the proposed algorithm can effectively solve the MTSOSP. The excellent performance of DRL-GA in large-scale problems reflects its ability to cope with actual scheduling scenarios. The

TABLE VI
BEST/AVE RESULTS OF RUNNING EACH ALGORITHM 30 TIMES

Scenario	DRL-GA	IGA [33]	KBGA [42]	DPABC [43]	ALNS-TI [34]	NS [44]
100-1	**755(710.9)**	590(551.53)-	604(562.67)-	595(562.57)-	591(559.33)-	581(546.73)-
100-2	**841(784.8)**	640(582.97)-	637(582.43)-	637(594.3)-	637(585.3)-	641(565.37)-
100-3	**777(729.77)**	593(560.1)-	591(559.33)-	580(562.2)-	614(559)-	627(551.67)-
200-1	**1368(1234.6)**	949(882.67)-	957(892.4)-	981(906.97)-	968(894.93)-	1027(865.57)-
200-2	**1427(1371.23)**	1107(1024.87)-	1060(1014.1)-	1137(1037.23)-	1058(1023.53)-	1081(1006.07)-
200-3	**1416(1339.63)**	1112(1033.33)-	1106(1031.2)-	1096(1031.57)-	1102(1025.53)-	1069(999.17)-
300-1	**1774(1677.9)**	1371(1326.27)-	1407(1338.83)-	1410(1345.73)-	1394(1338.3)-	1379(1303.8)-
300-2	**1399(1315.83)**	1110(1040.27)-	1116(1033.2)-	1124(1050.1)-	1110(1048.6)-	1113(1005.83)-
300-3	**1771(1675.1)**	1353(1294.57)-	1364(1303.97)-	1401(1317.1)-	1367(1306.2)-	1373(1273.1)-
400-1	**2597(2411.47)**	1906(1814.4)-	1904(1817.2)-	1944(1836.3)-	1894(1806.97)-	1878(1766.1)-
400-2	**2598(2396.77)**	1904(1828.17)-	1940(1829.17)-	1938(1873.97)-	2028(1852.43)-	1874(1802.67)-
400-3	**2405(2306.37)**	1842(1760.2)-	1840(1754.73)-	1904(1792.1)-	1833(1770.43)-	1838(1723.43)-
500-1	**2903(2787.23)**	2384(2249.43)-	2426(2258.77)-	2395(2305.37)-	2377(2257.57)-	2295(2205.47)-
500-2	**2841(2661.37)**	2250(2142.87)-	2264(2127.73)-	2303(2179.47)-	2314(2150.33)-	2199(2100.77)-
500-3	**2814(2609.53)**	2156(2057.2)-	2158(2068.87)-	2175(2115.33)-	2149(2069)-	2136(2039.9)-
600-1	**2842(2678.6)**	2289(2142.97)-	2232(2139.37)-	2331(2206.1)-	2287(2172.33)-	2271(2113.1)-
600-2	**3065(2967.4)**	2491(2363.63)-	2511(2361.7)-	2520(2433.73)-	2464(2375.4)-	2451(2317.97)-
600-3	**3317(3148.17)**	2665(2508.9)-	2617(2536.3)-	2650(2564.27)-	2692(2519.7)-	2544(2457.87)-
700-1	**3960(3713.23)**	3194(3046.67)-	3246(3069.23)-	3210(3118.8)-	3174(3066.53)-	3124(2992.23)-
700-2	**3729(3555.7)**	3068(2872.87)-	2986(2867.03)-	3037(2936.9)-	3055(2887)-	2936(2840.27)-
700-3	**3949(3792)**	3092(3007.6)-	3114(3019.8)-	3186(3078.13)-	3159(3024.23)-	3177(2979.43)-
800-1	**4160(3989.17)**	3244(3165.17)-	3264(3183.2)-	3394(3275.8)-	3349(3223.53)-	3319(3128.67)-
800-2	**4053(3805.97)**	3297(3074.4)-	3182(3098.5)-	3313(3149.23)-	3277(3103.43)-	3144(3029.2)-
800-3	**4463(4243.9)**	3558(3403.7)-	3578(3408.1)-	3636(3480.03)-	3603(3416)-	3539(3368.7)-
900-1	**4607(4413)**	3873(3735.5)-	3877(3727.5)-	3929(3809.8)-	3864(3748.87)-	3832(3667.37)-
900-2	**4378(4112.53)**	3540(3358.3)-	3484(3372.27)-	3612(3458.1)-	3554(3391.37)-	3479(3313.23)-
900-3	**4409(4215.2)**	3610(3450.83)-	3569(3442.47)-	3628(3501.83)-	3576(3449.67)-	3504(3359.43)-
1000-1	**4604(4350.7)**	3792(3640.6)-	3825(3652.13)-	3822(3735.2)-	3828(3666.1)-	3808(3600.53)-
1000-2	**4721(4536.03)**	4059(3915.03)-	4015(3909.03)-	4115(3986.03)-	4116(3948.13)-	3963(3828.77)-
1000-3	**5029(4668.33)**	4134(3850.67)-	4037(3866.17)-	4072(3935.57)-	4082(3868.23)-	3911(3758.67)-

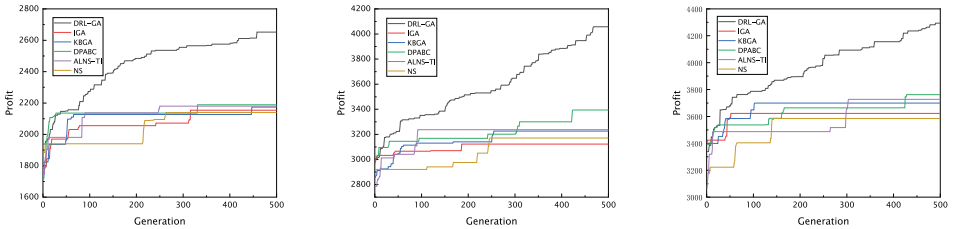

Fig. 5. Convergence Curves in 600, 800, and 1000 Task Scales of Scenarios. (a) 600 Task Scale. (b) 800 Task Scale. (c) 1000 Task Scale.

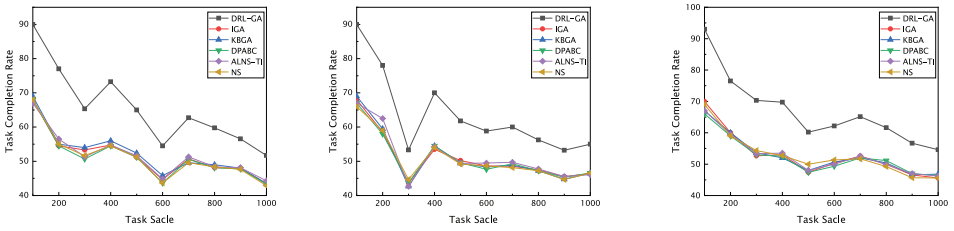

Fig. 6. Comparisons on task completion rate. (a) Results of each scenario numbered 1. (b) Results of each scenario numbered 2. (c) Results of each scenario numbered 3.

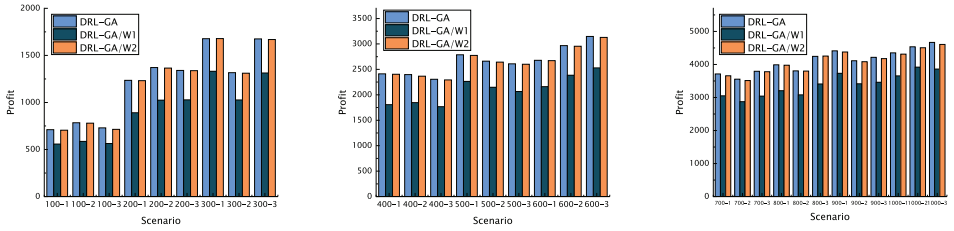

Fig. 7. Results of algorithms with different improvement strategies. (a) 100–300 task scale. (b) 400–600 task scale. (c) 700–1000 task scale.

indicator-based state space design ensures that the DRL model is highly generalizable and adaptable to different problem scenarios. DRL-GA can more easily exploit the advantages of GA population search than using DRL alone, and this solution construction method is important for improving GA performance.

DRL-GA presents an idea of combining the DRL method with EA, and we chose the classical GA to fully exploit

TABLE VII
CPU TIME OF EACH SCENARIO

Scenario	DRL-GA	DPABC [43]	ALNS-TI [34]	NS [44]
100-1	**15.10**	20.02	15.80	15.70
100-2	**12.80**	17.44	14.11	13.12
100-3	**13.19**	17.88	14.39	**13.19**
200-1	**24.06**	33.92	26.39	24.38
200-2	**25.56**	34.89	26.25	24.78
200-3	**25.65**	35.96	26.69	25.67
300-1	**36.94**	52.35	38.33	**36.94**
300-2	**37.02**	53.49	39.80	37.85
300-3	**39.55**	55.63	40.82	39.60
400-1	**65.16**	95.99	67.98	66.16
400-2	**61.88**	92.22	63.91	62.11
400-3	**65.81**	97.38	68.01	65.84
500-1	**87.58**	132.06	90.60	88.19
500-2	90.21	131.30	89.31	**86.16**
500-3	**85.12**	125.92	86.46	86.09
600-1	105.70	156.48	106.60	**103.63**
600-2	**104.19**	154.08	105.64	104.85
600-3	**104.07**	155.94	107.70	104.63
700-1	**152.03**	229.40	153.71	152.66
700-2	**141.95**	215.39	145.30	142.41
700-3	151.00	228.32	153.44	**150.75**
800-1	174.46	262.95	175.64	**173.02**
800-2	**170.63**	256.72	171.52	171.56
800-3	**179.41**	269.65	180.09	179.42
900-1	**203.68**	309.26	204.75	204.87
900-2	**196.02**	298.69	198.92	196.49
900-3	**203.10**	307.75	204.69	203.74
1000-1	**220.56**	334.56	223.19	221.23
1000-2	230.06	352.00	233.72	**229.82**
1000-3	**233.41**	353.18	235.89	236.46

the advantage of GA's strong global search ability. The EA combined with DRL can also choose other algorithms, such as ant colony algorithm, particle swarm algorithm, etc. The choice of the algorithm combination should be considered in conjunction with the problem characteristics and EA search process.

VI. CONCLUSION

Multitype satellite observations allow for the full exploitation of the respective strengths of different types of satellites, making the best use of resources while ensuring adequate observational profits for the task. In this work, we innovatively combined DRL and EA for solving the MTSOSP. We used a method of generating initial and neighborhood search solutions using the Dueling DQN model in the GA framework, which effectively utilized the valid information in the problem to obtain high-quality solutions quickly. The enhanced strategy designed in the algorithm improved the algorithm's search performance while preventing it from falling into the local optimum. The simulation results show that DRL-GA outperforms the competitors in terms of solution profits and task completion rates. In addition, the excellent performance in large-scale scheduling scenarios demonstrates that DRL-GA has a significant potential to be applied to real satellite scheduling systems.

This work proposes a new approach to the MTSOSP. In the future, this problem will be investigated in depth from several perspectives. More complex situations require the study of robust task scheduling problems or online scheduling problems by considering the uncertainty of the observation task, equipment, environment, etc. Other reinforcement learning methods can also be applied to MTSOSPs. In addition, other forms of combining reinforcement learning with EAs could be worth exploring.

ACKNOWLEDGMENT

Special thanks to Prof. Ke Tang for his guidance in writing and revising this article.

REFERENCES

[1] 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.

[2] F. Xhafa and A. W. Ip, "Optimisation problems and resolution methods in satellite scheduling and space-craft operation: A survey," *Enterp. Inf. Syst.*, vol. 15, no. 8, pp. 1022–1045, 2021.

[3] C. Han, Y. Gu, G. Wu, and X. Wang, "Simulated annealing-based heuristic for multiple agile satellites scheduling under cloud coverage uncertainty," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 53, no. 5, pp. 2863–2874, May 2023.

[4] S.-A. Boukabara, J. Eyre, R. A. Anthes, K. Holmlund, K. M. S. Germain, and R. N. Hoffman, "The Earth-observing satellite constellation: A review from a meteorological perspective of a complex, interconnected global system with extensive applications," *IEEE Geosci. Remote Sens. Mag.*, vol. 9, no. 3, pp. 26–42, Sep. 2021.

[5] E. Chuvieco et al., "Historical background and current developments for mapping burned area from satellite earth observation," *Remote Sens. Environ.*, vol. 225, pp. 45–64, May 2019.

[6] P. Minnett et al., "Half a century of satellite remote sensing of sea-surface temperature," *Remote Sens. Environ.*, vol. 233, Nov. 2019, Art. no. 111366.

[7] S. Wang, L. Zhao, J. Cheng, J. Zhou, and Y. Wang, "Task scheduling and attitude planning for agile earth observation satellite with intensive tasks," *Aerosp. Sci. Technol.*, vol. 90, pp. 23–33, Jul. 2019.

[8] Y. He, G. Wu, Y. Chen, and W. Pedrycz, "A two-stage framework and reinforcement learning-based optimization algorithms for complex scheduling problems," 2021, *arXiv:2103.05847*.

[9] H. Kim and Y. K. Chang, "Mission scheduling optimization of SAR satellite constellation for minimizing system response time," *Aerosp. Sci. Technol.*, vol. 40, pp. 17–32, Jan. 2015.

[10] Y. Chen, D. Zhang, M. Zhou, and H. Zou, "Multi-satellite observation scheduling algorithm based on hybrid genetic particle swarm optimization," in *Advances in Information Technology and Industry Applications*. Berlin, Germany: Springer, 2012, pp. 441–448.

[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, May 2014.

[12] D. Zhang, L. Guo, B. Cai, N. Sun, and Q. Wang, "A hybrid discrete particle swarm optimization for satellite scheduling problem," in *Proc. IEEE Conf. Anthol.*, 2013, pp. 1–5.

[13] Z. Chang and Z. Zhou, "Three multi-objective memtic algorithms for observation scheduling problem of active-imaging AEOS," 2022, *arXiv:2207.01250*.

[14] X. Fatos, J. Sun, B. Admir, B. Alexander, and B. Leonard, "Genetic algorithms for satellite scheduling problems," *Mobile Inf. Syst.*, vol. 8, no. 4, pp. 351–377, 2012.

[15] 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.

[16] G. Povéda et al., "Evolutionary approaches to dynamic earth observation satellites mission planning under uncertainty," in *Proc. Genet. Evol. Comput. Conf.*, 2019, pp. 1302–1310.

[17] 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.

[18] K.-J. Zhu, J.-F. Li, and H.-X. Baoyin, "Satellite scheduling considering maximum observation coverage time and minimum orbital transfer fuel cost," *Acta Astronautica*, vol. 66, nos. 1–2, pp. 220–229, 2010.

[19] G. Wu, Q. Luo, X. Du, Y. Chen, P. N. Suganthan, and X. Wang, "Ensemble of metaheuristic and exact algorithm based on the divide-and-conquer framework for multisatellite observation scheduling," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 58, no. 5, pp. 4396–4408, Oct. 2022.

[20] X. Chen, G. Reinelt, G. Dai, and A. Spitz, "A mixed integer linear programming model for multi-satellite scheduling," *Eur. J. Oper. Res.*, vol. 275, no. 2, pp. 694–707, Jul. 2019.

[21] J. Berger, N. Lo, and M. Barkaoui, "QUEST—A new quadratic decision model for the multi-satellite scheduling problem," *Comput. Oper. Res.*, vol. 115, Mar. 2020, Art. no. 104822.

[22] 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, Sep. 2021.

[23] 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, Feb. 2020.
[24] Z. Chang, Z. Zhou, L. Xing, and F. Yao, "Integrated scheduling problem for earth observation satellites based on three modeling frameworks: An adaptive bi-objective memetic algorithm," *Memet. Comput.*, vol. 13, no. 2, pp. 203–226, 2021.
[25] L. Wei, L. Xing, Q. Wan, Y. Song, and Y. Chen, "A multi-objective memetic approach for time-dependent agile earth observation satellite scheduling problem," *Comput. Ind. Eng.*, vol. 159, Sep. 2021, Art. no. 107530.
[26] O. Vinyals, M. Fortunato, and N. Jaitly, "Pointer networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 28, 2015, pp. 1–9.
[27] Y. He, L. Xing, Y. Chen, W. Pedrycz, L. Wang, and G. Wu, "A generic Markov decision process model and reinforcement learning method for scheduling agile earth observation satellites," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 52, no. 3, pp. 1463–1474, Mar. 2022.
[28] Y. Tian, X. Li, H. Ma, X. Zhang, K. C. Tan, and Y. Jin, "Deep reinforcement learning based adaptive operator selection for evolutionary multi-objective optimization," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 7, no. 4, pp. 1051–1064, Aug. 2023.
[29] Y. Du, J.-Q. Li, X.-L. Chen, P.-Y. Duan, and Q.-K. Pan, "Knowledge-based reinforcement learning and estimation of distribution algorithm for flexible job shop scheduling problem," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 7, no. 4, pp. 1036–1050, Aug. 2023.
[30] J. Wang, E. Demeulemeester, and D. Qiu, "A pure proactive scheduling algorithm for multiple earth observation satellites under uncertainties of clouds," *Comput. Oper. Res.*, vol. 74, pp. 1–13, Oct. 2016.
[31] D. Wei, S. Li, X. Wu, Y. Song, and R. Tan, "Design of electromagnetic compatibility test platform for transformer fire-fighting nitrogen injection extinguishing system," in *Proc. IOP Conf. Ser. Earth Environ. Sci.*, vol. 687, 2021, pp. 1036–1050.
[32] Y. Kang, E. Jang, J. Im, and C. G. Kwon, "A deep learning model using geostationary satellite data for forest fire detection with reduced detection latency," *GISci. Remote Sens.*, vol. 59, no. 1, pp. 2019–2035, 2022.
[33] J. Zhang and L. Xing, "An improved genetic algorithm for the integrated satellite imaging and data transmission scheduling problem," *Comput. Oper. Res.*, vol. 139, Mar. 2022, Art. no. 105626.
[34] L. He, M. De Weerdt, and N. Yorke-Smith, "Time/sequence-dependent scheduling: The design and evaluation of a general purpose tabu-based adaptive large neighbourhood search algorithm," *J. Intell. Manuf.*, vol. 31, no. 4, pp. 1051–1078, 2020.
[35] C. G. Valicka et al., "Mixed-integer programming models for optimal constellation scheduling given cloud cover uncertainty," *Eur. J. Oper. Res.*, vol. 275, no. 2, pp. 431–445, Jun. 2019.
[36] Y. Gu, C. Han, Y. Chen, S. Liu, and X. Wang, "Large region targets observation scheduling by multiple satellites using resampling particle swarm optimization," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 59, no. 2, pp. 1800–1815, Apr. 2023.
[37] J. Wu et al., "Frequent pattern-based parallel search approach for time-dependent agile earth observation satellite scheduling," *Inf. Sci.*, vol. 636, Jul. 2023, Art. no. 118924.
[38] G. Wu, X. Mao, Y. Chen, X. Wang, W. Liao, and W. Pedrycz, "Coordinated scheduling of air and space observation resources via divide-and-conquer framework and iterative optimization," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 59, no. 4, pp. 3631–3642, Aug. 2023.
[39] E.-G. Talbi, "Machine learning into metaheuristics: A survey and taxonomy," *ACM Comput. Surveys*, vol. 54, no. 6, pp. 1–32, 2021.
[40] Z. Waiming, H. Xiaoxuan, X. Wei, and J. Peng, "A two-phase genetic annealing method for integrated earth observation satellite scheduling problems," *Soft Comput.*, vol. 23, no. 1, pp. 181–196, 2019.
[41] Z. Wang, T. Schaul, M. Hessel, H. Hasselt, M. Lanctot, and N. Freitas, "Dueling network architectures for deep reinforcement learning," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 1995–2003.
[42] Y. Song, L. Xing, M. Wang, Y. Yi, W. Xiang, and Z. Zhang, "A knowledge-based evolutionary algorithm for relay satellite system mission scheduling problem," *Comput. Ind. Eng.*, vol. 150, Dec. 2020, Art. no. 106830.
[43] X. Jiang, Y. Song, and L. Xing, "Dual-population artificial bee colony algorithm for joint observation satellite mission planning problem," *IEEE Access*, vol. 10, pp. 28911–28921, 2022.
[44] D. Pisinger and S. Ropke, "Large neighborhood search," *Handbook of Metaheuristics*. Cham, Switzerland: Springer, 2019, pp. 99–127.

Yanjie Song received the double B.S. degrees in management from Tianjin University, Tianjin, China, in 2017, and the Ph.D. degree in engineering from the National University of Defense Technology, Changsha, China, in 2023.

He is an Assistant Professor with National Defense University, Beijing, China. He has published more than 40 papers in IEEE TRANSACTIONS ON AEROSPACE AND ELECTRONIC SYSTEMS, *Swarm and Evolutionary Computation*, *Information Sciences*, *Journal of Management Studies*, *Expert Systems With Applications*, and other journals. He has authored one academic book, obtained seven National Invention Patents, and hosted/participated in more than 15 projects. His research interests include computational intelligence, evolutionary algorithm, combinatorial optimization, and deep reinforcement learning.

Dr. Song is currently the Guest Editor of the *Swarm and Evolutionary Computation* and a Reviewer of the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS, IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, IEEE TRANSACTIONS ON AEROSPACE AND ELECTRONIC SYSTEMS, *Swarm and Evolutionary Computation*, *Knowledge-Based Systems*, *Information Sciences*, *Expert Systems With Applications*, and more than 20 other journals.

Junwei Ou received the B.Sc. degree in computer science from the School of Computer and Science Information Engineering, Henan University, Kaifeng, China, in 2015, the M.Sc. degree in computer science from the College of Information Engineering, Xiangtan University, Xiangtan, China, in 2019, and the Ph.D. degree in Control Science and Engineering from the National University of Defense Technology, Changsha, China, in 2023.

He is currently a Lecturer with the Department of Computer Science and Cyberspace Security College, Xiangtan University. His research interests include evolutionary computation, multiobjective optimization, and resource scheduling.

Witold Pedrycz (Life Fellow, IEEE) is a Professor with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB, Canada. He is also with the Systems Research Institute, Polish Academy of Sciences, Warsaw, Poland. His main research directions involve computational intelligence, granular computing, and machine learning.

Prof. Pedrycz is a recipient of several awards, including Norbert Wiener Award from the IEEE Systems, Man, and Cybernetics Society, IEEE Canada Computer Engineering Medal, a Cajastur Prize for Soft Computing from the European Centre for Soft Computing, a Killam Prize, a Fuzzy Pioneer Award from the IEEE Computational Intelligence Society, and 2019 Meritorious Service Award from the IEEE Systems Man and Cybernetics Society. He serves as an Editor-in-Chief for *Information Sciences* and *WIREs Data Mining and Knowledge Discovery* (Wiley) and a Co-Editor-in-Chief of *International Journal of Granular Computing* (Springer) and *Journal of Data, Information, and Management* (Springer). He is a Foreign Member of the Polish Academy of Sciences and a Fellow of the Royal Society of Canada.

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

Since August 2022, he has been with the KINDI Centre for Computing Research, Qatar University, Doha, Qatar, as a Research Professor. His research interests include randomization-based learning methods, swarm and evolutionary algorithms, pattern recognition, deep learning and applications of swarm, and evolutionary and machine learning algorithms.

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 of the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, the IEEE TRANSACTIONS ON CYBERNETICS, *Information Sciences*, and *Pattern Recognition* and the Founding Co-Editor-in-Chief of *Swarm and Evolutionary Computation*.

Xinwei Wang (Member, IEEE) received the Ph.D. degree in engineering from Beihang University, Beijing, China, in 2019.

He is a Lecturer with the Queen Mary University of London (QMUL), London, U.K. He was a Postdoctoral Fellow with TU Delft, Delft, The Netherlands, from 2020 to 2022, and QMUL from 2019 to 2020, respectively. He has authored over 30 papers, including those in *Transportation Research—Part C: Emerging Technologies*, IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS, IEEE TRANSACTIONS ON FUZZY SYSTEMS, IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, IEEE TRANSACTIONS ON AEROSPACE AND ELECTRONIC SYSTEMS. Over the years, he has integrated computational intelligence, machine learning and systems engineering for risk assessment, motion planning, and decision making in intelligent systems.

Dr. Wang is a recipient of the Marie Sklodowska-Curie Actions Co-Fund Fellowship (2022), and IEEE ITSS Young Professionals Travelling Fellowship (2022). He is an Associate Editor of the *Journal Advances in Space Research*, and a member of Young Editorial Board of the *Astrodynamics*.

Lining Xing received the bachelor's degrees in economics and in science from Xi'an Jiaotong University, Xi'an, China, in 2002, and the Ph.D. degree in management science from the National University of Defense Technology, Changsha, China, in 2009.

He visited the School of Computer, University of Birmingham, Birmingham, U.K., from November 2007 to November 2008. He is a Professor with the School of Electronic Engineering, Xidian University, Xi'an. His research interests include intelligent optimization methods and scheduling of resources.

Yue Zhang received the B.S. degree in engineering management from Nanjing Agricultural University, Nanjing, China, in 2019. She is currently pursuing the integrated master's and Ph.D. degree with the School of Reliability and Systems Engineering, Beihang University, Beijing, China.

She received national funding in 2023 to pursue a joint Ph.D. program with the Department of Industrial and Systems Engineering, National University of Singapore, Singapore. Her research interests include reliability, task allocation, mathematical programming, and computational intelligence.