Swarm and Evolutionary Computation 77 (2023) 101233

Contents lists available at ScienceDirect

Swarm and Evolutionary Computation

journal homepage: www.elsevier.com/locate/swevo

Deep reinforcement learning method for satellite range scheduling problem

Junwei Ou [a], Lining Xing [a,*], Feng Yao [a,1], Mengjun Li [a], Jimin Lv [a], Yongming He [a], Yanjie Song [a], Jian Wu [a], Guoting Zhang [b]

[a] College of Systems Engineering, National University of Defense Technology, Changsha, 410073, Hunan, China
[b] Beijing Institute of Tracking and Telecommunications Technology, Beijing 100094, China

ARTICLE INFO

Keywords:
Satellite range scheduling problem (SRSP)
Deep reinforcement learning (DRL)
Decompose
Heuristic scheduling method

ABSTRACT

The satellite range scheduling problem (SRSP) is a range of combinatory optimization, which plays a vital role in the regular operation and mission accomplishment of in-orbit satellites. However, with the increase in the number of satellites and the client requirements, there is some limitation in dealing with the SRSP for existing methods, especially on large-scale problems. Therefore, we propose a deep reinforcement learning (DRL) method, which is integrated into a heuristic scheduling method for the satellite range scheduling problem. The core idea of the algorithm is to decompose the problem into two subproblems: (1) Assignment problem, which assigns each task on different antennas. (2) Single antenna scheduling problem, which determines the execution start and end time of selected tasks on the antenna. The two subproblems are performed iteratively and modeled as a general paradigm. In the paradigm, the DRL is to determine the process of task assignment, and the heuristic scheduling method can quickly solve the single antenna scheduling problem. The objective function of the scheduling problem is to maximize the total reward. The DRL updates the gradient information based on the reward obtained by the heuristic scheduling method. To verify this idea, various scale experiments are considered to examine the performance of training scenarios. Experimental results show that the proposed paradigm combining DRL with a heuristic scheduling method can effectively deal with the SRSP.

1. Introduction

The communications between satellites and ground stations have received increasing interest in recent years [1]. In particular, communication in military and civilian missions plays a significant root in a variety of services. These services, including weather predictions, surveillance, geodesy, and navigation, need to be provided by satellites [2–4]. However, with the increase of satellites and the limitation of ground stations, scheduling satellites to service efficiently is becoming a challenging issue.

The satellite range scheduling problem (SRSP) is the process of selecting appropriate ground station antennas and time windows [5] for satellites' communication. SRSP was proposed by the Air Force Institute of Technology in [6]. The SRSP composes a set of satellite requests, a set of ground station antennas, and the visibility time windows of each request-antenna pair [7]. If a scenario only includes a single antenna, the problem is defined by single-resource range scheduling; otherwise, it is called multi-resource range scheduling. In the paper, the visibility time windows are provided by customers. The communication between satellite requests and ground station antennas must be done within the range of visible windows. An antenna only supports a satellite request at a visible time window [8]. When an antenna is communicating with a satellite, it may or not allow for interruption due to a high-priority request. Furthermore, there are numerous constraints in the SRSP, which include a fixed turnaround time for supporting successive satellite requests.

The SRSP is a typical NP-hard combination problem [9,10], which schedules ground stations on satellites to satisfy some constraints efficiently. Moreover, with the increase in resources and the mission, settling large-scale SRSP and providing efficient customer service is becoming challenging. Thus, some deterministic algorithms [11,12] for SRSP were proposed. These deterministic algorithms can find a set of optimal solutions under simple constraints. However, multiple constraints and large scale of tasks and antennas make it difficult for these algorithms to find optimal solutions. The population-based algorithms can handle highly complex constraints and obtain a set of approximate solutions in a relatively short period. Compared with

* Corresponding author.
E-mail addresses: junweiou@163.com (J. Ou), xinglining@gmail.com (L. Xing), yaofeng@nudt.edu.cn (F. Yao), mjli11260744@sina.com (M. Li), zealousis@163.com (J. Lv), heyongming14@nudt.edu.cn (Y. He), songyj_2017@163.com (Y. Song), 1551699723@qq.com (J. Wu), zhanggt_beijing@126.com (G. Zhang).
1 Co-first authors.

https://doi.org/10.1016/j.swevo.2023.101233
Received 30 July 2022; Received in revised form 2 December 2022; Accepted 1 January 2023
Available online 11 January 2023

deterministic algorithms, population-based algorithms are effective in dealing with SRSP. Thence, these algorithms have been widely accepted as a tool for SRSP. But, providing fast and reliable solutions is still a challenging task for population-based algorithms. Therefore, it is necessary to design an efficient algorithm to deal with SRSP.

Deep reinforcement learning (DRL) has recently attracted much attention from researchers in the field of combinatorial optimization. Most of the research makes use of DRL to learn the searching rules, including the traveling salesman problem (TSP) [13], job-shop scheduling problem (JSP) [14], and vehicle routing problem (VRP) [15]. The model of DRL is trained by the trial-and-error process and can be regarded as a black-box heuristic, which is obtained by training the characteristic of the problem [16]. By the trained network, the optimal solutions can be obtained in a very short time. In addition, there is a strong generalization ability in the trained DRL model, which can solve the problem of unexplored characteristics [17].

In the study, SRSP is first decomposed into two subproblems, i.e., an assignment problem and a single antenna scheduling problem. Then, the DRL method determines tasks to be assigned to different antennas, while a heuristic scheduling method obtains a scheduling plan for each antenna. Hence, each antenna can solve the SRSP by a range of subproblems. The main contributions of this paper are summarized as follows:

(1) A general paradigm composed of a DRL and a heuristic scheduling method is developed to deal with the SRSP. In the paradigm, the SRSP is modeled into two subproblems solved by two different technologies.
(2) In order to learn the characteristics of the satellite mission, the DRL-based method is modeled and decides the process of task assignment.
(3) In order to verify the performance of the proposed algorithm, we consider some scenarios. Experimental results show that the proposed paradigm can outperform other algorithms in dealing with the SRSP.

This paper is organized as follows. Section 2 reviews existing studies on the SRSP and technologies. Section 3 provides a preliminary for the SRSP. Section 4 presents the proposed algorithm in detail. Section 5 gives experimental results and a comparison of the algorithm to other algorithms. Finally, a conclusion is shown in Section 6.

2. Literature review

SRSP is a class of combinatorial optimization problems that have received much attention in recent years. The research of SRSP is of great importance in major national projects and national defense construction. An increasing amount of optimization methods are considered to solve the SRSP. The optimization methods for SRSP can be divided into three categories: deterministic algorithms, population-based algorithms, and machine learning algorithms. In this section, we mainly review the three methods.

(1) $deterministic\ algorithms$: The deterministic algorithms can guarantee the optimal solution under certain assumptions by mathematical programming approaches, such as branch and bound [12,18], tree search [19,20], dynamic programming [21,22]. When the scale of the problem is large, the exact algorithm can provide a solution to the problem on the one hand and an initial solution for the heuristic method; on the other hand, a better solution can be searched. Due to the high computational cost of the exact algorithm, it is difficult to use in large-scale practical problems. For example, Barulescu et al. [23] proposed a heuristic method for actual SRSP. Luo et al. [5] proposed a combination of relaxation and heuristics to obtain a more accurate upper bound for the SRSP. Marinelli et al. [24] developed a Lagrangian version of the Fix-and-Relax MIP heuristic. Chen et al. [25] used an improved adaptive large neighborhood search algorithm for SRSP. Although it has a better result according to the experimental analysis, the cost of the method is time-consuming [5]. The solution time of the exact solution algorithm for large-scale problems is difficult for users to accept [24,26].

(2) $population-based\ algorithms$: The population-based algorithms can obtain an approximation solution by meta-heuristic approaches, such as genetic algorithm (GA) [27,28], ant colony optimization (ACO) [29,30], and particle swarm optimization (PSO) [31]. These heuristic approaches, which work with a population of solutions, have been widely accepted as a major tool for solving SRSP. Compared with the deterministic algorithms, these meta-heuristic approaches are able to obtain a set of approximation solutions at a lower computational cost. The research of these meta-heuristic approaches [27,29] has been considered promising to solve SRSP. For example, Deb et al. [32] proposed a pareto-based evolutionary algorithm for improving the population convergence and diversity by a non-dominated sorting and a selection strategy, respectively. The Pareto-based evolutionary algorithm has been applied to various areas including the combinatorial optimization problem. Song et al. [33] proposed a multi-objective optimization for the SRSP based on the STK simulation. Li et al. [34] proposed an improved genetic algorithm by using site coding for SRSP. Du et al. [35] proposed a general multi-objective optimization evolutionary algorithms (MOEA) based memetic algorithm framework for multi-objective SRSP. Song et al. [36] proposed an improved genetic algorithm with the neighborhood search for SRSP. In [33–36], they use a population-based evolutionary algorithm to optimize the SRSP for obtaining a set of solutions. The research mainly designs some variation operators (e.g., crossover and mutation) to reproduce new representations. Zhang et al. [37] proposed a two-stage update pheromone ACO algorithm to solve the satellite resource scheduling problem. Peng et al. [30] considered a remote satellite scheduling problem and solved it by ACO. Zhang et al. [29] proposed an improved ACO with the bounds, update, and initialization of pheromones to deal with the SRSP. Chai et al. [38] proposed a double ACO for scheduling the imaging and the data transmission requests by designing an independent set model.

(3) $machine\ learning\ algorithms$: Machine learning (ML) is one of the most salient techniques in many applications, and provides a powerful tool for extracting useful and hidden patterns from a big dataset. Many ML-based meta-heuristics achieve high-quality results in solving various complex optimization problems. Research in applying ML to solve combinatorial optimization problems has become increasingly popular. ML can be divided into two categories: unsupervised learning and supervised learning. The most common ML tasks include classification, regression, and clustering. Classification and regression are supervised ML task that needs a set of labeled data, while clustering is unsupervised learning that does not require labeled data. In the combinatorial optimization community, it is difficult for us to find a set of optimal solutions as labels for training a model. Hence, we often use unsupervised learning, such as DRL, to deal with combinatorial optimization problems. DRL has been employed to address SRSP due to the success of DRL in solving TSP [13], VRP [39,40]. Although the training of DRL needs to consume much time, obtained the trained neural network model can directly provide high-quality solutions in a relatively short time. For example, Vinyals et al. [41] proposed a pointer network model, which is a supervised learning model to solve TSP. Since the supervised learning model requires a large number of labels, it was extended to DRL [13,42]. Li et al. [43] proposed a DRL method integrating self-attention in both the encoder and decoder for heterogeneous capacitated VRP. He et al. [44] proposed a finite Markov decision process model based on a constructive heuristic algorithm for satellite scheduling problems. Wei et al. [45] proposed a deep reinforcement learning and parameter transfer-based approach for satellite scheduling problem in a non-iterative manner. An encoder–decoder structure neural network is applied to the deep reinforcement learning procedure for producing a high-quality solution.

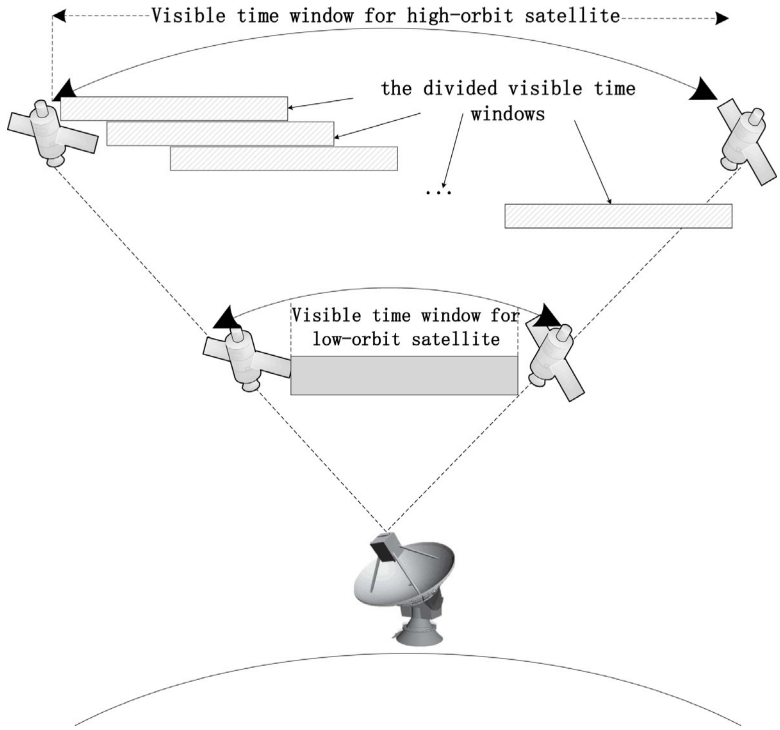

Fig. 1. Illustration of SRSP.

3. Preliminaries

In this section, we mainly introduce the SRSP description, the mathematical model of the SRSP, model process using the form of DRL.

3.1. Problem description

The SRSP refers to the process of satellite data reception and command annotation after the satellite and the ground station establish a communication link when the satellite flies over the ground station. During a period, the satellite orbits the earth to reach the ground station multiple times to form various visibility time windows. The goal of SRSP is to establish a satellite-to-ground station antenna connection within a visible time window. The visible time windows need to be assigned to different request tasks to satisfy the satellite requests. As shown in Fig. 1, two satellites can connect with a ground station, but a ground station antenna can only establish a connection with one satellite at a time. Hence, the main goal of SRSP is to allocate visible time windows to satellites reasonably.

3.2. Mathematical model

For the SRSP, a satellite mission consists of a set of $Tasks$ and a set of $Timewindows$, and then each task is allocated a visible $Timewindow$. **Data** :

Let A be the set of ground antennas. Let R be the set of satellite tasks. There are N tasks in total. For each task $r \in R$, it can be presented by the tuple of (p_r, o_r, a_r, $b_{r,a}$, $e_{r,a}$, s_r, t_r).

(a) p_r: the profit of task $r \in R$; There is a bigger value of p_r indicating that the profit of task r is higher.
(b) o_r: the service of task $r \in R$; A task can perform different services, such as tracking, uplink, reception and ranging session.
(c) a_r: the antennas supporting task $r \in R$; A task can be supported by several antennas.
(d) $b_{r,a}$: the beginning time of the task $r \in R$ on visibility window for antennas $a \in a_r$.
(e) $e_{r,a}$: the ending time of visibility window on the request $r \in R$ for antennas $a \in a_r$.
(f) s_r: the service time of the task $r \in R$.
(g) t_r: the turnaround time of the request $r \in R$.

In addition, s_a denotes that the antenna $a \in A$ can support several services. **Variables** :

Let $x_{r,a}$ be 1 when request $r \in R$ on antenna $a \in A$ is scheduled, otherwise $x_{r,a}$=0.

Let $y_{r,a}$ present the start time of request $r \in R$ when the request r is performed in the antenna $a \in a_r$. **Objective** :

The SRSP requests not only the maximum number of scheduled tasks but also the maximization of the profit of the scheduled task. In practice, when a task $i \in R$ on the antenna j is scheduled, x_{ij} will be set 1. Otherwise, it will be set 0. In order to optimize the problem, the mathematical model is defined as follows:

$$\textbf{Max} \sum_{i=1}^{N} \sum_{j=1}^{M} x_{ij} p_i, \tag{1}$$

where M represents the number of antenna. **Constraints** :

$$\sum_{a \in a_r} x_{r,a} \leq 1 \quad (r \in R), \tag{2}$$

$$b_{r,a} \leq y_{r,a} \leq e_{r,a} \quad (r \in R; a \in a_r), \tag{3}$$

$$y_{r,a} + d_r + t_r \leq y_{q,a} \quad (r, q \in R), \tag{4}$$

$$s_r \subset s_a \quad (r \in R; a \in A), \tag{5}$$

The constraint (2) indicates a task can be allocated to at most one antenna. The constraint (3) indicates each task must be performed within the available time windows at a ground station. The constraint (4) indicates an antenna cannot overlap more than a request at the same time. The constraint (5) indicates the antenna must support the service of the task.

Furthermore, some assumptions need to be done for developing analysis.

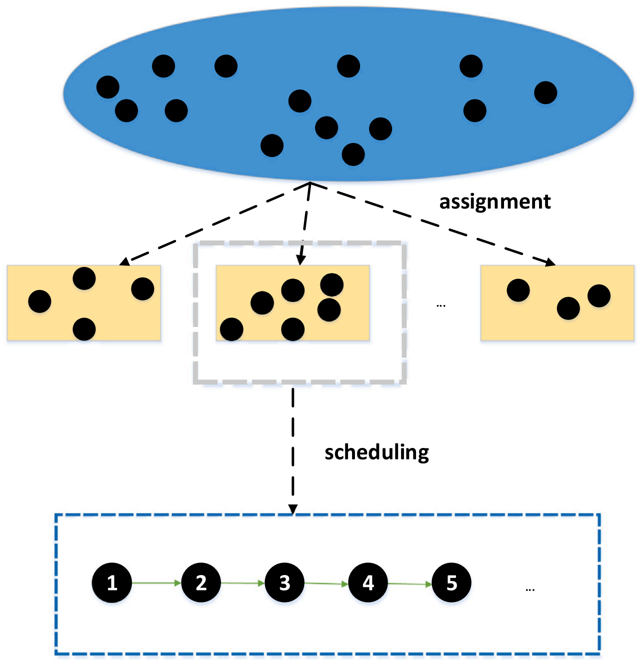

Fig. 2. A schematic diagram of the satellite range scheduling.

1. The shift time of t_r is a constant.
2. The task cannot stop once it is performed.
3. The time frame we plan is within the scope of the determination.

3.3. Modeling process

The process of solving the SRSP can be realized in two parts. First, each task in all requests is allocated an antenna according to the antenna's support relationship. The process can be viewed as a task assignment. Then, all tasks in each antenna are set to the visiting order according to the beginning time of the task. By the two processes, tasks can be determined based on the antenna being performed and its start and end times. The diagram of scheduling SRSP is presented in Fig. 2. Each satellite task is assigned to a specific antenna using the DRL-based method. For the satellite missions on an antenna, the tasks are ranked based on their profit values, and a scheduling plan can be obtained by eliminating possible conflicts. The scheduling plan is introduced in Section 4.4. For the process of the task assignment, we use DRL to guide the selection. The DRL has been applied to many combinatorial optimization problems, such as TSP [13] and VRP [43]. In this paper, we model the process of the task assignment as a Markov decision process (MDP) defined by 4-tuple $M = \{S, A, \tau, r\}$. Importantly, the state space S, the action space A, the state transition rule τ, and the reward function r are introduced as follows:

State : In our method, the state $s_t \in S$ contains two different states, which are the task state RT_t, and the antenna An_t. For the antenna state An_t, it describes the situation of antennas at the ith time step. The state An_t is expressed as $An_t = \{an_t^1, an_t^2, \ldots, an_t^m\} = \{(o_t^1, G_t^1), (o_t^2, G_t^2), \ldots, (o_t^m, G_t^m)\}$, where o_t^i represents the relationship of the antenna to support request tasks and it does not change over time. G_t^i represents the tasks to be visited by the antenna and it will change over time. The task state RT_t is expressed as $RT_t = \{RT_t^1, RT_t^2, \ldots, RT_t^m\} = \{(p_t^1, b_t^1, e_t^1, s_t^1, ta_t^1), \ldots, (p_t^m, b_t^m, e_t^m, s_t^m, ta_t^m)\}$, where p_t^1, b_t^1, e_t^1, s_t^1 and ta_t^1 indicate the profit, the beginning time, the ending time, the service time and the turnaround time, respectively.

Action : The action in the method is to select a task and an antenna, which can be regarded as selecting an antenna for a task. The action $a_t \in A$ is treated as (r_t^i, an_t^j), that is the task r^i will be assigned to the an_j.

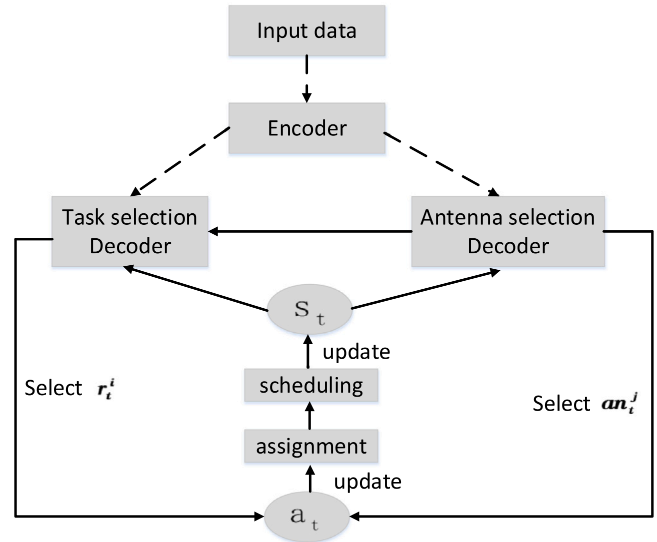

Fig. 3. The illustration of the general framework.

Transition : By the transition rule τ, the next state s_{t+1} is obtained based on the actions a_t as well as previous state s_t, that is, $s_{t+1} = \tau(s_t)$. The state RT_{t+1} is updated as follows:

$$G_{t+1}^k = \begin{cases} [G_t^k, rt^j] & \text{if } k = j, \\ [G_t^k, g_t^j] & \text{otherwise,} \end{cases} \tag{6}$$

Reward : The reward is used to compute the profit of tasks. The maximum reward represents a better result. Then, the reward in steps t and $t+1$ is defined as follows:

$$r_{t+1} = r(s_{t+1}, a_{t+1}) = pr_i, \tag{7}$$

where pr_i is the profit of selecting the task i.

4. The proposed algorithm

In the section, we use the DRL-based method to assign tasks to different antennas. We select an attention-based neural network for the DRL to train the policy. Moreover, we introduce the training process and a constructive heuristic algorithm.

4.1. General framework

The selected DRL-based method is to learn a stochastic policy $\pi_\theta(a_t|s_t)$, which minimizes the loss objective. To do this, we use a deep neural network to train the parameter θ. First, the initial state s_0 represents the initial stage where tasks will be assigned to the antennas. Then, we use the π to form a task assignment scheme until the terminate state θ_τ is satisfied. Thus, the probability chain rule is calculated as follows:

$$p(s_\tau|s_0) = \prod_{t=0}^{\tau-1} \pi_\theta(a_t|s_t) p(s_{t+1}|s_t, a_t), \tag{8}$$

where τ is the number of all steps.

In order to elaborate on the general framework, we provide a simple example to illustrate the process in Fig. 3. First, the input data, including tasks and antennas, is encoded into a feature space for better representation with raw features in the first step (t=0). Its output is a vector that remains unchanged over time. Then, the policy selects an antenna (an^j) from all the antennas according to the antenna decoder selection. Next, based on the antenna index identified by the antenna decoder selection, the policy selects the appropriate tasks that the antenna can support via the task decoder selection. After the mission assignment, we will do a mission scheduling for the mission on that antenna. The constructive heuristic algorithm we chose for the scheduling is presented in Section 4.4. Finally, the state s_t is updated based on the action a_t constructed by the selected antenna an^j and task r^i. By the way, we can determine the assignment scheme of all tasks on which the tasks of an antenna are performed and generate a scheduling plan for using users.

4.2. Architecture of policy network

We follow an encoder–decoder structure for the architecture, as shown in Fig. 3. The structure includes an encoder and two decoders. The encoder of our method uses a multi-decoder attention model [46] composed of a task selection decoder and an antenna selection decoder. Afterward, we are concerned with the detail of the encoder, task selection decoder, and antenna selection decoder, respectively.

Encoder : Since the features in the task need to be extended rational and have broader characteristics. The encoder is used to enrich the information for the antenna selection decoder. First, the features of each task, that is, $\tilde{x}^i = (o^i, p^i, b^i, e^i, s^i, t^i)$, are mapped into a higher-dimensional space using a linear projection ($d_h = 128$ in the paper). Then, the Transformer-style model based on the multi-head attention (MHA) mechanism and a feedforward (FF) sublayer is followed for better feature extraction. By the way, the h_0^i linearly projected by $\tilde{x}^i$ is encoded into h_N^i based on N attention layers. In the paper, the multi-head attention layers are defined as follows:

$$Q_{l,y}, K_{l,y}, V_{l,y} = h_l W_{l,y}^Q, h_l W_{l,y}^K, h_l W_{l,y}^V, \tag{9}$$

$$Z_{l,y} = softmax\left(\frac{Q_{l,y}K_{l,y}^T}{\sqrt{dim_k}}\right)V_{l,y}, \tag{10}$$

$$\begin{aligned} MHA(h_l) &= MHA(h_l W_l^Q, h_l W_l^K, h_l W_l^V) \\ &= Concat(Z_{l,1}, Z_{l,2}, \dots, Z_{l,Y})W_l^O, \end{aligned} \tag{11}$$

where $h_l = (h_l^0, h_l^1, \dots, h_l^n)$ is the dim-dimensional embedding for the task; W_l^Q, $W_l^K \in \mathbb{R}^{Y\times dim\times dim_k}$, $W_l^V \in \mathbb{R}^{Y\times dim\times dim_v}$ and $W_l^O \in \mathbb{R}^{dim\times dim_k}$ are learnable parameters in layer l; $dim_k = (dim/Y)$ is the query/key dimension and $dim_v = (dim/Y)$ is the value dimension; $dim = 128$ is the dimension in a high-dimensional space; Y=8 is the number of heads in the attention.

Then, the following embedding h_{l+1} is obtained by the two strategies, which are consisted of the skip-connection and a batch normalization (BN) layer. The core of the two strategies is defined as follows:

$$r_l^i = BN(h_l^i + MHA^i(h_l)), \tag{12}$$

$$h^i(l+1) = BN(r_l^i + FF(r_l^i)), \tag{13}$$

Antenna selection Decoder : The role of the antenna selection decoder is to select an antenna according to the probability distribution implemented by two embeddings: antenna feature embedding and task feature embedding.

First, the antenna features context C_t^V at step t is defined as follows:

$$C_t^V = [G_{t-1}], \tag{14}$$

where G_{t-1} denotes the supporting tasks by the current antenna. To fully extract the antenna feature context, it needs to be linearly projected with the parameters W_1 and b_1. Then it is integrated into the next feature embedding H_t by the 512-dim FF layer with the ReLU activation function. The process is calculated as follows:

$$H_t^V = FF(W_1 C_t^V + b_1), \tag{15}$$

Then, the task feature context C_t^E at time t is defined as follows:

$$C_t^E = [h_t^1, h_t^2, \dots, h_t^m], \tag{16}$$

The feature needs to extract more context in a similar way as mentioned above: the linear projection with trainable parameters W_2 and b_2 and a 512-dim FF layer. The way can be calculated as follows:

$$H_t^E = FF(W_2 C_t^E + b_2). \tag{17}$$

Moreover, the two features, including the antenna feature embedding H_t^V and task feature embedding C_t^E are merged into a new feature that is linearly projected by the parameter W_3 and b_3. Then, the probability of selecting the next antenna at time step t is calculated as follows:

$$H_t = W_3[H_t^V, H_t^E] + b_3, \tag{18}$$

$$p_t = softmax(H_t) \tag{19}$$

where the softmax function is used to normalize H_t. Note that the method of retrieving the maximum probability greedily selects the next antenna.

Task selection Decoder : After selecting an antenna by the antenna selection decoder, the task selection decoder considers these tasks that can be supported by the selected antenna. We aim to learn a probability distribution $\overline{p}_t$ to assign the corresponding task that is to be visited by the antenna. First, we defined a feature context H_t^c composed of feature vector $\hat{h}_n$ and current task embedding h_{t-1}^i as follows:

$$H_t^c = [\hat{h}_N, h_{t-1}^i], \tag{20}$$

where $\hat{h}_N = \frac{1}{n}\sum_{i\in X} h_N^i$. Then, the context H_t^c and the embedding h_N are fed into an MHA layer. The MHA layer is similar to the one mentioned above for better feature extraction and is defined as follows:

$$\hat{H}_t^c = MHA(H_t^c W_c^Q, h_N W_c^K, h_N W_c^V) \tag{21}$$

where W_c^Q, W_c^K and W_c^V are training parameters. Then, we compute the probability of all tasks at time t as follows:

$$u_t = C \cdot tanh\left(\frac{q_t^T k_t}{\sqrt{dim_k}}\right), \tag{22}$$

where $q_t = \hat{H}_t^c W_{comp}^Q$ and $k_t = h_N W_{comp}^K$ are training parameters and C is set to 10. As a result, the probability of selecting a task for the antenna at time t is calculated as follows:

$$\overline{p}_t = softmax(u_t) \tag{23}$$

4.3. Training process

Algorithm 1 proposed Algorithm

Input:
initialized parameters θ;
The set of positive samples for current batch, P_n;
The set of unlabeled samples for current batch, U_n;
Ensemble of classifiers on former batches, E_{n-1};

```
1:  for each iter = 1, 2, ... do
2:    generate N problem instances randomly;
3:    for each i = 1, 2, ..., M do
4:      a batch b = N_i
5:      while not terminated do
6:        select an action a_{t,b} ~ π_θ(a_{t,b}|s_{t,b}) ;
7:        obtain a reward r_{t,b} and update next state s_{t+1,b};
8:      end while
9:      R_b = Σ_{t=0}^{τ} r_{t,b};
10:     GreedyRollout with baseline υ_φ and compute its reward R_b^{BL};
```

11: $d_\theta \leftarrow -\frac{1}{B}\sum_{b=1}^{B}(R_b - R_b^{BL})\nabla \log \pi_\theta\ (s_{\tau,b}|s_{0,b})$;
12: $\theta \leftarrow \mathrm{Adam}(\theta, d_\theta)$;
13: **end for**
14: **if** OneSidedPairedTTest(π_θ, v_ϕ) $< \alpha$ **then**
15: $\phi \leftarrow \theta$
16: **end if**
17: **end for**

The model is trained using the policy gradient method similar to [43], as presented in Algorithm 1. Using the technique consists of two networks. First, a policy gradient network π_θ is used to select an action and give the probability for the antennas and tasks; Then, a baseline network v_ϕ is a greedy roll-out baseline, which evaluates the reward by selecting the antenna and task with maximum probability. During the training, we randomly initialize some parameters and generate N problem instances. For each instance, we compute the reward using the policy gradient in line 9. Moreover, R_b^{BL} is an expected reward conducted by a greedy roll-out of the baseline network in line 10. In order to update the parameters of the baseline network, a paired t-test on several instances is employed to verify the significant differences between the two parameters in line 15.

4.4. Constructive heuristic algorithm

Once the tasks are assigned to the antennas, a heuristic algorithm is needed to determine the order in which the tasks are performed on the antennas, and to determine the start and end times of the tasks. In the paper, we use an approach based on task profit sorting to deal with a single antenna scheduling problem. The pseudo-code of the heuristic algorithm is shown as Algorithm 2. The heuristic algorithm first sorts all the tasks to be planned according to the task profit. This method allows the tasks with the largest profit to be executed first, resulting in a better profit. If the task satisfies the constraints, we choose the earliest allowed start time for the task. Then, the task is then placed into the planned set. With this heuristic algorithm, we can determine the execution order and time window of the tasks.

Algorithm 2 heuristic algorithm

Input:
A sequence of tasks T;
Output:
$plan$;
Sort T by the task profit in descending order.
for each task in T **do**
if the task t_i satisfy all constraints **then**
Schedule tasks on the earliest possible time window.
$plan \leftarrow plan \cup t_i$.
end if
end for

5. Experimental design and analysis

The experimental scenarios in the paper follow a series of Chinese satellites. Then, we use some algorithms to assess the performance of the proposed method.

5.1. Test instance and satellite orbital parameters

We design some test problems with task sizes of 50, 100, 150, and 200. Each task scheduling period is 24 h. After pretreatment, each task can be served on one or multiple antennas with one or more visible time windows. All satellite tasks are randomly generated based on actual conditions. In this way, the performance of the algorithm can be better tested in combination with actual cases. For the task profit, we consider using an integer to describe the importance of the task that the user provides. In the scenarios, we use an integer ranging from 1 to 10 to assign the profit value for each task.

Table 1
Satellite orbital parameters.

Parameter	LSA	E	I	AP	RAAN	MA
Value	7141701.7	0.000627	98.5964	95.5069	342.307	125.2658

The tasks in the experiment were all generated from low earth orbit (LEO) satellites. The satellite orbit parameters include the length of the semi-major axis (LSA), eccentricity (E), inclination (I), the argument of perigee (AP), right ascension of the ascending node (RAAN), and mean anomaly (MA). The initial orbital parameters for the satellite are presented in Table 1.

In the experiment, the data for all satellite tasks were generated based on ground station operational practices. In order to explain in detail how to generate the test instance, we provide a procedure referring to [5] as follows:

step 1 Initialize the problem parameters.
step 2 Generate n satellite tasks one by one.

- step 2–1 Set the turnaround time of the ith task to 10 min, t_r, and randomly generate a number from 10 to 20 as the service of rth task, s_r.
- step 2–2 Randomly generate a number from 1 to 3 as the number of visible antennas of the ith task, m_i. Randomly select m_i visible antennas from m antennas as the visible antennas of ith task.
- step 2–3 Randomly generate m_r integers in the interval [1, 1440 - t_r - s_r] as the beginning time of visible time windows between ith task and its visibility antennas. The ending time of visibility windows can easily be determined by the start time plus the service time.
- step 2–4 Randomly generate an integer from 1 to 10 as the task profit.
- step 2–5 Return the Step 2–1 to generate the next task until all tasks are completed.

5.2. Parameter settings and compared algorithm

First, the training instances are generated by random sampling, and the size of each iteration is 12800, and 256 batches are set for each iteration. In the experiment, we follow 30 iterations for all problem sizes to verify the effectiveness of our method, although more iterations can achieve better results. Then, the features of the tasks and antennas are encoded into a 128-dimensional high-dimensional space, and the dimension of the hidden layer is also set to 128-dimensional. In addition, we select the Adam optimizer to train the policy parameters with an initial learning rate 10^{-4} and decaying 0.995 per iteration for convergence.

To verify the performance, we select five classical heuristic methods for comparison algorithms, including (1) GA [34], a basic genetic algorithm for combinatorial optimization problems; (2) LS [47], a construction heuristic algorithm based on the local search for solving SRSP; (3)tabu search (TS) [48], an efficient heuristic method for solving the combinatorial optimization problem. (4)knowledge-based genetic algorithm (KBGA) [49], a knowledge-based evolutionary algorithm for relay satellite system mission scheduling problem. (5)IRICGA [50], an individual reconfiguration based integer coding genetic algorithm for multi-satellites imaging scheduling problem. Furthermore, we adjusted the objectives and related settings of all contrasting algorithms so that they share the same objective as the proposed ones.

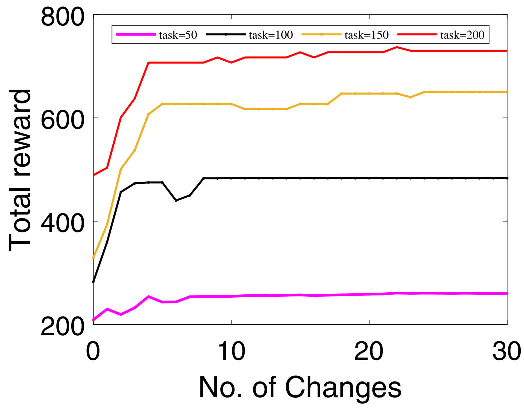

Fig. 4. Total reward of the first 30 episodes in different scenarios.

Table 2
CUP time spent on the training model and running.

Scenarios	Training CUP time	Run CUP time
task=50	37 h 29 min 40 s	4 min 29 s
task=100	130 h 52 min 2 s	14 min 16 s
task=150	275 h 52 min 53 s	30 min 26 s
task=200	520 h 37 min 37 s	52 min 22 s

5.3. Performance of applying DRL to SRSP

(1) *Convergence Analysis*: In this section, we mainly analyze the algorithms' performance on different task scales. First, we provide the curve of total reward over the first 30 episodes in different scenarios in Fig. 4.

It can be seen from Fig. 4 that the DRL algorithm can obtain better results on different task scales after being trained, implying that it can have many advantages in dealing with SRSP. In the early stage, the total reward of each episode rises sharply, indicating that the proposed algorithm can quickly respond to different scenarios in a shorter period of time. In the latter stages, the proposed algorithm can make use of the training information to achieve good stability. There is a possible reason for the result. The proposed DRL-based algorithm can deal with task allocation well, and a heuristic algorithm is used to find better solutions. The proposed DRL-based algorithm is integrated into the heuristic algorithm to enhance its search ability. As a result, the proposed algorithm can effectively deal with the SRSP.

(2) *Training Time Analysis*: The section mainly discusses the training time of the model. Table 2 presents the training time and the running time obtained by the proposed DRL model in the different scenarios. As shown in Table 2, the model training takes a lot of time on different task scales. This means that the DRL-based method needs to spend much time on the training of the model. At the same time, the heuristic algorithm in the DRL also consumes some time. However, after model training, the algorithm can obtain a set of excellent solutions in a short time. Therefore, although the model training takes more time, a scheduling scheme can be quickly obtained once the model training is completed. In conclusion, it is acceptable to spend a lot of time training the model to get a schedule quickly.

5.4. Analysis of algorithm performance

This section investigates the efficiency of our DRL-based method against other methods. We provide a comparative experiment about the average reward of the six algorithms, where the best values obtained by one of six algorithms are highlighted in bold face. It can be seen from Table 3 that the RDL-based algorithm outperforms other algorithms on the majority of the task scales. However, it performs slightly worse than GA, KBGA, and IRICGA for task = 200. For all problems of all scales, LS fails to show very promising performance. In addition, as the size of the task increases, the performance difference between the proposed algorithm and other algorithms is not significant. Clearly, there is only a marginal difference between all algorithms. Furthermore, the proposed DRL-based algorithm is slightly worse than other algorithms except for LS, when the task is 200. One possible explanation is that the selected heuristic method is not suitable for dealing with a single antenna scheduling problem. This is understandable because the heuristics used are relatively simple. Overall, the proposed DRL-based method has more advantages in dealing with SRSP than other algorithms.

Table 3
The average reward of the six algorithms.

Scenarios	DRL	GA	LS	TS	KBGA	IRICGA
task=50	**260.1**	254.3	183	251.3	255.9	243.1
task=100	**483.2**	443.2	259.1	482.4	456.4	453.3
task=150	**650.5**	578.1	367.5	634.3	615.2	598.2
task=200	730.2	738.8	458.0	714.6	**812.9**	768.6

Table 4
Average Rankings of the algorithms (Friedman).

MIGD	Average ranking value	Final rank
DRL	1.7532	1
GA	3.9654	5
LS	5.8989	6
TS	3.7500	4
KBGA	2.5233	2
IRICGA	3.1010	3

5.5. Study in the Friedman test

To verify the effectiveness of the proposed method, the Friedman test [51,52] is used to investigate further significance between different results at the 0.05 significance level. Table 4 presents the statistical results of the Friedman test among the six algorithms under comparison according to the profit value of the scheduled task. As shown in Table 4, the proposed DRL-based algorithm ranks first in SRSP among the six algorithms. The results of the Friedman test show that the proposed DRL-based algorithm is a robust model in this scenario. Moreover, the proposed DRL-based algorithm can produce more promising solutions than other algorithms in the metric value obtained by the Friedman test. This shows that the proposed DRL-based algorithm is more efficient than other methods. In total, the RDL-based algorithm significantly outperforms the other algorithms by a clear margin in terms of the Friedman test. This means that the DRL-based algorithm may be helpful for handling SRSP.

5.6. Discuss

5.6.1. Study of different task scale

To examine the effect of task scale on algorithms' performance, experiments are carried out on SRSP with different task scales, and task scales are set to 75, 125, and 175, which present small, moderate, and big scales, respectively. Experimental results of three algorithms in different scenarios are presented in Fig. 5. It can be observed from Fig. 5 that the profit of the task increases with the size of the task. There is a significant difference in the results between the three algorithms on different task scales. When the task size is set to 175, the gap between the algorithms is relatively small. The reason is that as the size of the task increases, the number of related tasks increases. This can lead to a high failure rate of the task in big-scale tasks.

Furthermore, we adopt another performance metric to investigate algorithms' performance on different task scales. The performance metric is called the rate of total profit, which is proportional to the

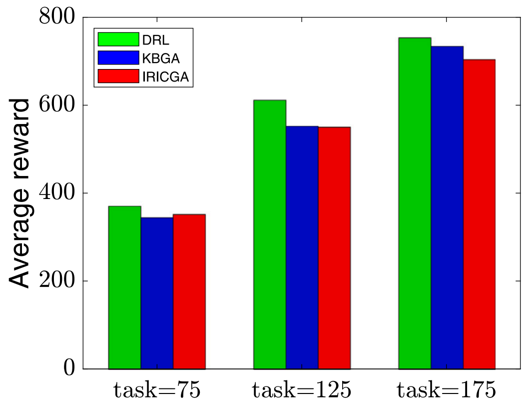

Fig. 5. Average reward of three algorithms in different scenarios.

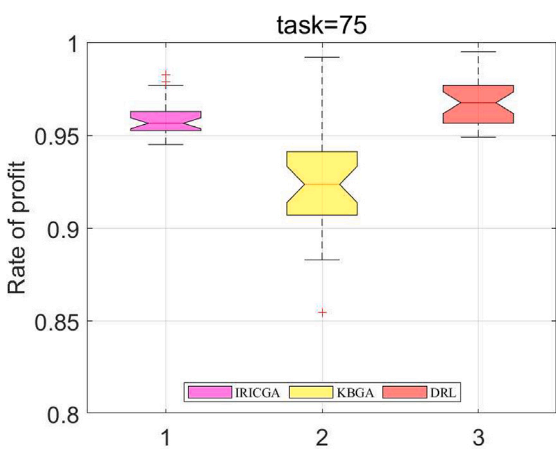

Fig. 6. Profit rate of three algorithms under 75 task scale.

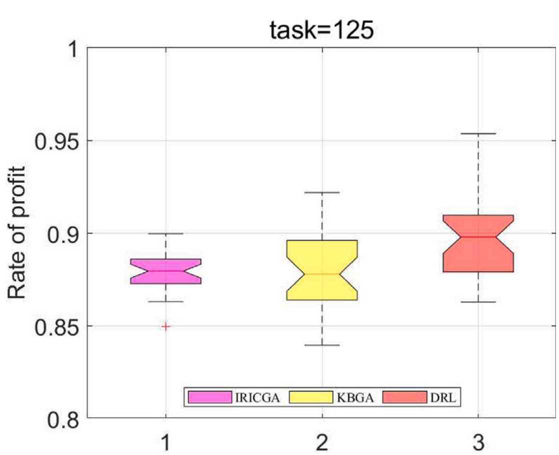

Fig. 7. Profit rate of three algorithms under 125 task scale.

total profit in the same task set. The rate of total profit is defined as follows:

$$\mathrm{pr} = \frac{\mathrm{tpst}}{\mathrm{tptts}} \times 100\%, \tag{24}$$

where *tpst* denotes the total profit of all scheduled tasks, and *tptts* denotes total profit of all tasks in the task set.

The obtained results in three different scenarios about *pr* metric are presented in Figs. 6, 7, 8. It can be seen from Fig. 6 that the rate of total profit in the three algorithms is more than 90%. When the task

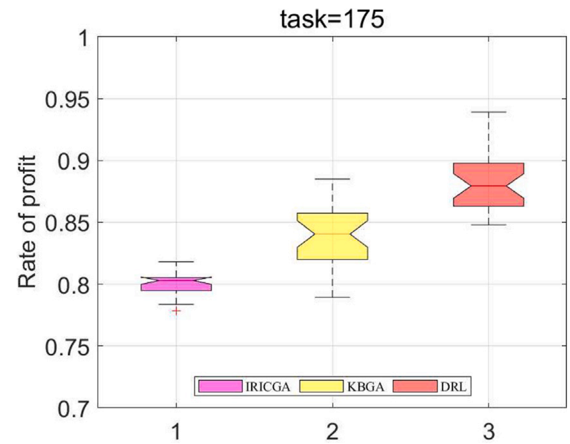

Fig. 8. Profit rate of three algorithms under 175 task scale.

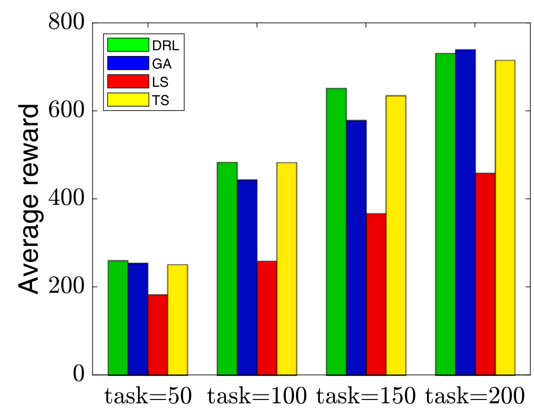

Fig. 9. Average reward of six algorithms in different scenarios.

size increases, the rate of total profit in three algorithms decreases. In addition, DRL significantly outperforms the other algorithms in terms of the *pr* metric. However, the gap in the *pr* metric becomes smaller when the task size increase. This means that all algorithms will face big challenges with the task size growing. In inclusion, DRL seems suitable for dealing with SRSP.

5.6.2. More discuss

In order to discuss the average reward of six algorithms in different scenarios, we provide an average reward of six algorithms in different scenarios in Fig. 9. It can be seen from Fig. 9 that the proposed DRL-based algorithm performs significantly better than other algorithms except for GA on the task = 200. Besides, the proposed DRL algorithm shows some appealing results on small-scale instances, implying the DRL-based method may be helpful for dealing with SRSP. However, when the task size is 200, the DRL-based method does not show promising results, and there is not much difference between them. This means the DRL-based algorithm has not much advantage in dealing with large-scale SRSP.

6. Conclusion

In the paper, we propose a deep reinforcement learning (DRL) method to deal with SRSP. According to the framework of DRL-based, the SRSP can be decomposed into two subproblems: the assignment problem and the single antenna scheduling problem, respectively. The proposed DRL-based algorithm solves the assignment problem, and the heuristic algorithm deals with the single antenna scheduling problem. It is concluded from the experimental comparison and analysis that the combination of DRL-based and the heuristic algorithm can effectively deal with SRSP.

Although the DRL-based method produces encouraging performance on SRSP, it needs to be examined in a broader of SRSP, including large-scale instances. Therefore, our future work will mainly study large-scale SRSP. In addition, we are considering using more machine learning methods, including DQN and transfer learning for SRSP. The reason is that the ML-based method can indeed improve the adaptability and solution performance of the algorithm.

CRediT authorship contribution statement

Junwei Ou: Conceptualization, Methodology, Software, Writing – original draft, Writing – review & editing. **Lining Xing:** Investigation, Funding acquisition. **Feng Yao:** Funding acquisition. **Mengjun Li:** Validation, Formal analysis, Supervision. **Jimin Lv:** Project administration, Resources. **Yongming He:** Validation, Investigation. **Yanjie Song:** Investigation. **Jian Wu:** Investigation. **Guoting Zhang:** Data Curation.

Declaration of competing interest

No author associated with this paper has disclosed any potential or pertinent conflicts which may be perceived to have impending conflict with this work. For full disclosure statements refer to https://doi.org/10.1016/j.swevo.2023.101233.

Data availability

No data was used for the research described in the article.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 72271240, 72201273, 71901213; in part by the Technical Field Foundation in 173 Program of National Defense Technology, China under Grant 2021-JCJQ-JJ-0049; in part by the Hunan Natural Science Foundation, China under Grant 2022JJ30671; and in part by the Hunan Postgraduate Research Innovation Project, China under Grand CX20220002.

References

[1] E. Escobar, M. Diaz, J.C. Zagal, Evolutionary design of a satellite thermal control system: Real experiments for a CubeSat mission, Appl. Therm. Eng. 105 (2016) 490–500.
[2] J. García-Pintado, J.C. Neal, D.C. Mason, S.L. Dance, P.D. Bates, Scheduling satellite-based SAR acquisition for sequential assimilation of water level observations into flood modelling, J. Hydrol. 495 (2013) 252–266.
[3] S. Spangelo, J. Cutler, Optimization of single-satellite operational schedules towards enhanced communication capacity, in: AIAA Guidance, Navigation, and Control Conference, 2012, p. 4610.
[4] K.-J. Zhu, J.-F. Li, H.-X. Baoyin, Satellite scheduling considering maximum observation coverage time and minimum orbital transfer fuel cost, Acta Astronaut. 66 (1–2) (2010) 220–229.
[5] K. Luo, H. Wang, Y. Li, Q. Li, High-performance technique for satellite range scheduling, Comput. Oper. Res. 85 (2017) 12–21.
[6] T.D. Gooley, Automating the Satellite Range Scheduling Process, Tech. Rep., Air Force Inst of Tech Wright-Pattersonafb Oh School of Engineering, 1993.
[7] Z. Zheng, J. Guo, E. Gill, Swarm satellite mission scheduling & planning using hybrid dynamic mutation genetic algorithm, Acta Astronaut. 137 (2017) 243–253.
[8] B. Wille, M.T. Wörle, C. Lenzen, VAMOS–verification of autonomous mission planning on-board a spacecraft, IFAC Proc. Vol. 46 (19) (2013) 382–387.
[9] A.J. Vazquez, R.S. Erwin, On the tractability of satellite range scheduling, Optim. Lett. 9 (2) (2015) 311–327.
[10] L. Barbulescu, J.-P. Watson, L.D. Whitley, A.E. Howe, Scheduling space–ground communications for the air force satellite control network, J. Sched. 7 (1) (2004) 7–34.
[11] C.A. Rigo, L.O. Seman, E. Camponogara, E. Morsch Filho, E.A. Bezerra, P. Munari, A branch-and-price algorithm for nanosatellite task scheduling to improve mission quality-of-service, European J. Oper. Res. (2022).
[12] X. Chu, Y. Chen, Y. Tan, An anytime branch and bound algorithm for agile earth observation satellite onboard scheduling, Adv. Space Res. 60 (9) (2017) 2077–2090.
[13] I. Bello, H. Pham, Q.V. Le, M. Norouzi, S. Bengio, Neural combinatorial optimization with reinforcement learning, 2016, arXiv preprint arXiv:1611.09940.
[14] Y. Wei, M. Zhao, A reinforcement learning-based approach to dynamic job-shop scheduling, Acta Automat. Sinica 31 (5) (2005) 765.
[15] W. Qin, Z. Zhuang, Z. Huang, H. Huang, A novel reinforcement learning-based hyper-heuristic for heterogeneous vehicle routing problem, Comput. Ind. Eng. 156 (2021) 107252.
[16] N. Brown, B. Arguello, L. Nozick, N. Xu, A heuristic approach to satellite range scheduling with bounds using Lagrangian relaxation, IEEE Syst. J. 12 (4) (2018) 3828–3836.
[17] L. Xin, W. Song, Z. Cao, J. Zhang, Multi-decoder attention model with embedding glimpse for solving vehicle routing problems, in: Proceedings of 35th AAAI Conference on Artificial Intelligence, 2021, pp. 12042–12049.
[18] X. Chu, Y. Chen, L. Xing, A branch and bound algorithm for agile earth observation satellite scheduling, Discrete Dyn. Nat. Soc. 2017 (2017).
[19] G. Beaumet, G. Verfaillie, M.-C. Charmeau, Feasibility of autonomous decision making on board an agile earth-observing satellite, Comput. Intell. 27 (1) (2011) 123–139.
[20] N. Zufferey, M. Vasquez, A generalized consistent neighborhood search for satellite range scheduling problems, RAIRO-Oper. Res. 49 (1) (2015) 99–121.
[21] Z. Liu, Z. Feng, Z. Ren, Route-reduction-based dynamic programming for large-scale satellite range scheduling problem, Eng. Optim. 51 (11) (2019) 1944–1964.
[22] H. Ren-Jie, L. Ju-Fang, C. Ying-Wu, Satellite orbit task merging problem and its dynamic programming algorithm, Syst. Eng. Electron. 31 (7) (2009) 1738–1742.
[23] L. Barbulescu, A. Howe, D. Whitley, AFSCN scheduling: How the problem and solution have evolved, Math. Comput. Modelling 43 (9–10) (2006) 1023–1037.
[24] F. Marinelli, S. Nocella, F. Rossi, S. Smriglio, A Lagrangian heuristic for satellite range scheduling with resource constraints, Comput. Oper. Res. 38 (11) (2011) 1572–1583.
[25] Y. Chen, Y. Song, Y. Du, M. Wang, R. Zong, C. Gong, A knowledge-based scheduling method for multi-satellite range system, in: International Conference on Knowledge Science, Engineering and Management, Springer, 2020, pp. 388–396.
[26] N. Zufferey, P. Amstutz, P. Giaccari, Graph colouring approaches for a satellite range scheduling problem, J. Sched. 11 (4) (2008) 263–277.
[27] M. Chen, J. Wen, Y.-J. Song, L.-n. Xing, Y.-w. Chen, A population perturbation and elimination strategy based genetic algorithm for multi-satellite TT&C scheduling problem, Swarm Evol. Comput. 65 (2021) 100912.
[28] J. Zhang, L. Xing, An improved genetic algorithm for the integrated satellite imaging and data transmission scheduling problem, Comput. Oper. Res. 139 (2022) 105626.
[29] Z. Zhang, F. Hu, N. Zhang, Ant colony algorithm for satellite control resource scheduling problem, Appl. Intell. 48 (10) (2018) 3295–3305.
[30] P. Gao, Y.J. Tan, J.F. Li, R.J. He, An ant colony algorithm for remote satellite and ground integration scheduling problem in parallel environment, in: Advanced Materials Research, vol. 791, Trans Tech Publ, 2013, pp. 1341–1346.
[31] Y. Chen, D. Zhang, M. Zhou, H. Zou, Multi-satellite observation scheduling algorithm based on hybrid genetic particle swarm optimization, in: Advances in Information Technology and Industry Applications, Springer, 2012, pp. 441–448.
[32] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Trans. Evol. Comput. 6 (2) (2002) 182–197.
[33] Y.-J. Song, X. Ma, X.-J. Li, L.-N. Xing, P. Wang, Learning-guided nondominated sorting genetic algorithm II for multi-objective satellite range scheduling problem, Swarm Evol. Comput. 49 (2019) 194–205.
[34] Y. Li, R. Wang, Y. Liu, M. Xu, Satellite range scheduling with the priority constraint: An improved genetic algorithm using a station ID encoding method, Chin. J. Aeronaut. 28 (3) (2015) 789–803.
[35] Y. Du, L. Xing, J. Zhang, Y. Chen, Y. He, MOEA based memetic algorithms for multi-objective satellite range scheduling problem, Swarm Evol. Comput. 50 (2019) 100576.
[36] Y.-J. Song, Z.-S. Zhang, B.-Y. Song, Y.-W. Chen, Improved genetic algorithm with local search for satellite range scheduling system and its application in environmental monitoring, Sustain. Comput.: Inform. Syst. 21 (2019) 19–27.
[37] Z. Zhang, N. Zhang, Z. Feng, Multi-satellite control resource scheduling based on ant colony optimization, Expert Syst. Appl. 41 (6) (2014) 2816–2823.
[38] W. Chai, C. Zhang, Double ant colony algorithm of multi-atellite mission planning based on graph theory, Radio Eng. 49 (6) (2019) 534–539.
[39] X. Chen, Y. Tian, Learning to perform local rewriting for combinatorial optimization, Adv. Neural Inf. Process. Syst. 32 (2019).
[40] J.M. Vera, A.G. Abad, Deep reinforcement learning for routing a heterogeneous fleet of vehicles, in: 2019 IEEE Latin American Conference on Computational Intelligence, LA-CCI, IEEE, 2019, pp. 1–6.
[41] O. Vinyals, M. Fortunato, N. Jaitly, Pointer networks, Adv. Neural Inf. Process. Syst. 28 (2015).
[42] M. Nazari, A. Oroojlooy, L. Snyder, M. Takác, Reinforcement learning for solving the vehicle routing problem, Adv. Neural Inf. Process. Syst. 31 (2018).

[43] J. Li, Y. Ma, R. Gao, Z. Cao, A. Lim, W. Song, J. Zhang, Deep reinforcement learning for solving the heterogeneous capacitated vehicle routing problem, IEEE Trans. Cybern. (2021).

[44] Y. He, L. Xing, Y. Chen, W. Pedrycz, L. Wang, G. Wu, A generic Markov decision process model and reinforcement learning method for scheduling agile earth observation satellites, IEEE Trans. Syst., Man, Cybern.: Syst. (2020).

[45] L. Wei, Y. Chen, M. Chen, Y. Chen, Deep reinforcement learning and parameter transfer based approach for the multi-objective agile earth observation satellite scheduling problem, Appl. Soft Comput. 110 (2021) 107607.

[46] J. Li, L. Xin, Z. Cao, A. Lim, W. Song, J. Zhang, Heterogeneous attentions for solving pickup and delivery problem via deep reinforcement learning, IEEE Trans. Intell. Transp. Syst. 23 (3) (2021) 2306–2315.

[47] L. Barbulescu, A.E. Howe, J.-P. Watson, L.D. Whitley, Satellite range scheduling: A comparison of genetic, heuristic and local search, in: International Conference on Parallel Problem Solving from Nature, Springer, 2002, pp. 611–620.

[48] O. Bräysy, M. Gendreau, Tabu search heuristics for the vehicle routing problem with time windows, Top 10 (2) (2002) 211–237.

[49] Y. Song, L. Xing, M. Wang, Y. Yi, W. Xiang, Z. Zhang, A knowledge-based evolutionary algorithm for relay satellite system mission scheduling problem, Comput. Ind. Eng. 150 (2020) 106830.

[50] E. Zhibo, R. Shi, L. Gan, H. Baoyin, J. Li, Multi-satellites imaging scheduling using individual reconfiguration based integer coding genetic algorithm, Acta Astronaut. 178 (2021) 645–657.

[51] J. Derrac, S. García, D. Molina, F. Herrera, A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms, Swarm Evol. Comput. 1 (1) (2011) 3–18.

[52] E. Osaba, E. Villar-Rodriguez, J. Del Ser, A.J. Nebro, D. Molina, A. LaTorre, P.N. Suganthan, C.A.C. Coello, F. Herrera, A tutorial on the design, experimentation and application of metaheuristic algorithms to real-world optimization problems, Swarm Evol. Comput. 64 (2021) 100888.