Clustering-Scheduling Methods for Oversubscribed Short-Term Tasks of Astronomical Satellites

YIN Xiaodan[1,2], *BAI Meng*[1,2*], *LI Zhuoheng*[1]

1. Laboratory of Satellite Operations Technology, Space Science Mission Operations Center, National Space Science Center, Chinese Academy of Sciences, Beijing 100190, P. R. China;
2. University of Chinese Academy of Sciences, Beijing 100049, P. R. China

(Received 30 November 2022; revised 20 February 2023; accepted 5 June 2023)

Abstract: When the observation requirement from users exceeds the satellite's observation capability, astronomy satellite task scheduling becomes an oversubscription problem. For the oversubscribed task scheduling of astronomical satellites, a framework with a clustering phase and a short-term task scheduling phase is designed. First, a task clustering model is established to reduce the size of the oversubscribed task. Second, using the clustered results as input, we develop a mathematical model of short-term scheduling for the tasks. Finally, we propose an improved artificial bee colony algorithm with adaptive hybrid search strategies (DirectABC). It introduces an adaptive elite global-local search strategy and an adaptive variable neighborhood optimal search strategy to the basic artificial bee colony algorithm (BasicABC). The proposed algorithm demonstrates superior optimum-searching capability and a faster convergence speed in the simulation. In addition, it effectively reduces the number of tasks in the clustering phase and improves task completion in the short-term task scheduling phase.

Key words: astronomy satellite task scheduling; oversubscription problem; task clustering; short-term task scheduling; artificial bee colony algorithm

CLC number: V474 **Document code:** A **Article ID:** 1005-1120(2023)03-0307-16

0 Introduction

Task scheduling is a crucial technology that every astronomical observation satellite mission will face. Astronomical satellite task scheduling typically becomes an oversubscription problem, i. e., the time and resources of a satellite are insufficient to fulfill all observation requests one by one[1], as researchers place more observational demands to discover novel laws and phenomena. According to ESA AO-10 documents, the Suzaku satellite's total available time during the AO-10 period was 5 951 ks, while the operation center received 8 330 ks of proposals. It indicates that the oversubscription fraction is 140%. Many missions face the problem of oversubscription[2], which presents an even more significant challenge for task scheduling techniques.

In astronomy satellite task scheduling, some research has been done, and some results have been obtained. Johnston et al.[3] designed the SPIKE system for the Hubble telescope. The system first generates a solution that satisfies most constraints, then iteratively repairs the plan by relocating or removing observations that do not satisfy the constraints. NASA and JAXA subsequently used the system in many missions. For NASA's SWIFT mission, Penn State University has developed a scheduling system similar to SPIKE. The system continues to view the scheduling problem as a constraint satisfaction problem and generates a daily schedule with up to 125 tasks[4]. ESA developed the XMAS system for the XMM-Newton mission, which uses algo-

*Corresponding author, E-mail address: baimeng@nssc.ac.cn.

How to cite this article: YIN Xiaodan, BAI Meng, LI Zhuoheng. Clustering-scheduling methods for oversubscribed short-term tasks of astronomical satellites[J]. Transactions of Nanjing University of Aeronautics and Astronautics, 2023, 40(3): 307-322.

http://dx.doi.org/10.16356/j.1005-1120.2023.03.007

rithms such as hill climbing, forbidden search, and simulated annealing[5]. The MrSPOCK system utilized by the Mars Express mission is based on a multi-objective genetic algorithm. The AIMS system was used by the INTEGRAL mission[6]. The Advanced Planning & Scheduling Initiative (APSI) developed the generic constraint processing platform that AIMS and MrSPOCK used[7]. EChO[8-9] employed a two-phase strategy, in which scientific requests are first planned by the genetic algorithm, followed by the insertion of as many operational tasks as possible to fill in the gaps in the plan. Huang et al.[10] developed a multi-objective genetic algorithm for the HXMT mission's long-term scheduling. In cooperation with China and France, Jaubert and Li et al.[11-12] developed a multi-objective genetic algorithm for rescheduling the SVOM mission. Wu et al.[13] designed a multi-objective observation mission planning algorithm (MOMPA) that considers the constraints of both satellite observation and data downlink. Liu et al.[14] developed a mathematical model for task scheduling of space astronomy satellites and designed a multi-objective genetic algorithm based on observation window sequences. She et al.[15] presented a method for calculating the optimal slew path for the SVOM mission based on the Primal-dual interior point algorithm. Xu et al.[16-17] developed a planning method based on tiling coverage rules for targets of opportunity.

Few of the above studies examined task scheduling in the context of oversubscription. A novel approach to solving the oversubscription problem is clustering. Recent astronomical satellites with large field-of-view offer the possibility of observing multiple tasks simultaneously. By clustering targets within the same field-of-view, an astronomical satellite can observe multiple targets simultaneously and fulfill more observation requests. For instance, the wide-field x-ray telescope (WXT) of the Einstein Probe satellite has a field-of-view of 3 600 deg^2. To this end, this paper examines clustering and scheduling techniques in the context of oversubscription.

Clustering has been studied less in astronomical satellite task scheduling, but it has been addressed in some research on Earth observation satellite (EOS) task scheduling. For instance, Zhao et al.[18] clustered densely distributed target points before using the Tabu algorithm to generate the local observation path. Before assigning tasks to multiple satellites, Du et al.[19] also used task clustering. Besides, in EOS task scheduling, She et al.[20] presented a new Agile Earth Observation Satellite mission planning algorithm based on the Modified Mixed-Integer Linear Programming (MILP) approach. Du et al.[21] developed an observation path planning algorithm based on the dynamic imaging mode and the grid discretization approach using the modified ant colony algorithm with sensational and consciousness strategy for area targets.

Inspired by the above studies, this paper proposes a four-phase framework to address the oversubscription of tasks, as shown in Fig.1. Our research focuses on the task clustering phase and the short-term scheduling phase. First, the relevant clustering constraints are analyzed in the task clustering phase to formulate a clustering model and cluster of the oversubscribed tasks to reduce the problem scale. Second, the constraints of resources, user requirements, and space environment are investigated in the short-term scheduling phase, and a mathematical model is then developed to maximize task completion and task priority. Finally, the artificial bee colony (ABC) algorithm is used to solve the clustering and scheduling problems.

To speed up the convergence and improve the global searching capability, a modified artificial bee colony algorithm is put forward by introducing the adaptive elite global-local search strategy to the employed bee phase and the adaptive variable neighborhood optimal search strategy to the onlooker bee phase, respectively. After experimental validation, the improved ABC algorithm exhibits excellent convergence and optimum-searching capabilities in both clustering and scheduling. This study offers theoretical and technical guidance for short-term task scheduling of astronomical satellites in the context of oversubscription.

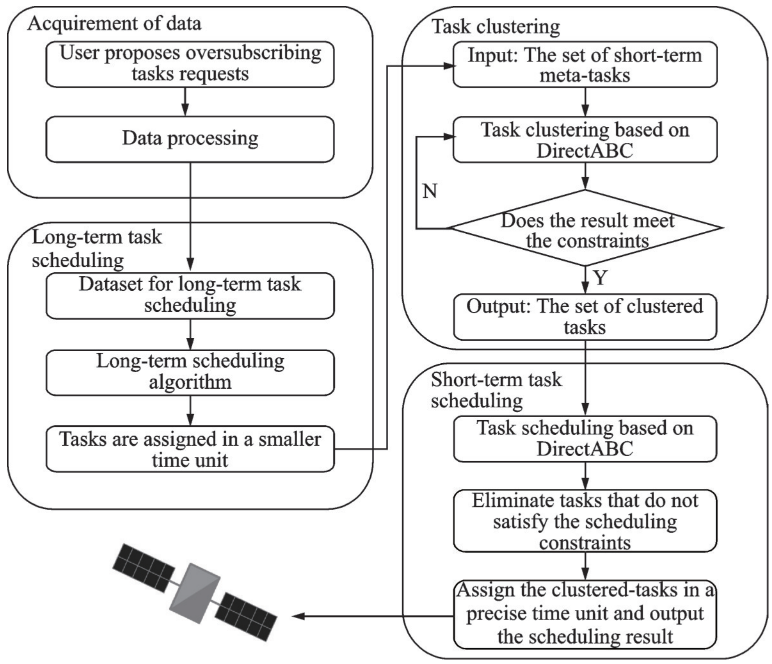

Fig.1 Four-phase framework for oversubscribed task scheduling

1 Problem Formulation

The section divides the model into two stages, clustering and short-term scheduling, and formulates the corresponding models.

1.1 Clustering problem for oversubscribed tasks

In actual missions, astronomers' requests typically exceed the satellite's capacity. For instance, the total duration of requests exceeds the available observation time of a satellite[1-2], resulting in an oversubscription problem.

Clustering is one possible solution to this problem. In machine learning, clustering means dividing the objects into several groups based on their inherent similarity[22]. Clustering for astronomical observation tasks has a similar definition. Tasks that satisfy certain constraints are clustered together for simultaneous completion. This method can be advantageous for mitigating oversubscription pressure, increasing observation efficiency, and maximizing scientific output.

1.1.1 Definition and description of the symbols

$M=\{m_i|i=1,2,\cdots,I\}$ denotes the set of meta-tasks, in which m_i is the ith meta-task.

$\boldsymbol{P}=[p_1,p_2,\cdots,p_I]$ is the vector of priorities of the meta-tasks. For example, p_i is the priority of the meta-task m_i.

$\boldsymbol{U}=[u_1,u_2,\cdots,u_I]$ is the vector of required observation durations of the meta-tasks. For example, u_i is the required duration of m_i.

$\boldsymbol{R}=[r_1,r_2,\cdots,r_I]$ is the vector of the right ascensions of the meta-tasks, in which r_i is the right ascension of meta-task m_i.

$\boldsymbol{D}=[d_1,d_2,\cdots,d_I]$ is the vector of the declinations of the meta-tasks, where d_i is the declination of m_i.

$W_i=\bigcup_{j=1}^{n_i} w_{i,j}$ is the visible time window of meta-task m_i, containing n_i sub-windows. $w_{i,j}=[w_{i,j}^{\mathrm{s}},w_{i,j}^{\mathrm{e}}]\cap \mathbf{Z}$ represents the jth time window of m_i. $w_{i,j}^{\mathrm{s}}$ is the start time of this window in minutes, and $w_{i,j}^{\mathrm{e}}$ is the end time in minutes.

$M'=\{S_k\}_{k=1}^{N}$ denotes the set of clustered tasks. $|M'|=N$, $|\cdot|$ represents the cardinal number of a set. $S_k=\{m_i\}_{i=n_1^k}^{n_{|S_k|}^k}\in M'$, is a clustered task that contains several meta-tasks. n_j^k is the index of jth meta-task in S_k.

$W_k'=\bigcap_{i=n_1^k}^{n_{|S_k|}^k} W_i$ is the visible time window of clustered task S_k. It takes the intersection of the set of all meta-tasks' visible time windows within the cluster. $W_k'=\bigcup_{j=1}^{n_k'} w_{k,j}'$ consists of n_k' time windows.

$\boldsymbol{P}'=[p'_1, p'_2, \cdots, p'_N]$, where p'_k represents the priority of the clustered task S_k, $k=1, 2, \cdots, N$.

$\boldsymbol{U}'=[u'_1, u'_2, \cdots, u'_N]$ represents the required duration of clustered tasks in the set of M'.

C_l^k denotes whether the clustered task S_k satisfies the lth constraint.

1.1.2 Constraints for clustering

(1) All meta-tasks in each clustered task must meet the following constraints.

① Time window

For any clustered task $S_k=\{m_i\}_{i=n_1^k}^{n_{|S_k|}^k}\in M'$, its visible time window set $W'_k=\bigcup_{j=1}^{n'_k} w'_{k,j}$ should satisfy the observation requirements of all meta-tasks within the class, i.e.

$$\forall i, \exists j: w'^{\mathrm{e}}_{k,j} - w'^{\mathrm{s}}_{k,j} \geqslant \max(u_i) \tag{1}$$

where $i=n_1^k, n_2^k, \cdots, n_{|S_k|}^k$; $j=1, 2, \cdots, n'_k$; $k=1, 2, \cdots, N$.

If the clustered task S_k satisfies this constraint, then $C_1^k=1$, otherwise $C_1^k=0$.

② Field-of-view of the payload

All meta-tasks in each clustered task should be visible to the payload. It indicates that the maximum angle between these meta-tasks and the payload must be less than the payload's field-of-view angle $v°$, as shown in Fig.2, i.e.

$$\forall i, j: \langle m_i, m_j \rangle \leqslant v \tag{2}$$

where $\langle \cdot, \cdot \rangle$ represents the angle between two tasks and the payload. Eq.(2) is equation to Eq.(3).

$$\forall i, j: \cos\langle m_i, m_j \rangle \geqslant \cos v \tag{3}$$

A meta-task in a 3D coordinate system is depicted as Fig.3. Assuming that the payload's position can be roughly regarded as the origin of the equatorial coordinate system, the three-dimensional unit vector of the meta-task m_i is $\boldsymbol{e}_i=(\cos r_i \cos d_i, \sin r_i \cdot \cos d_i, \sin d_i)$. Thus, meta-tasks m_i and m_j have the following relationship

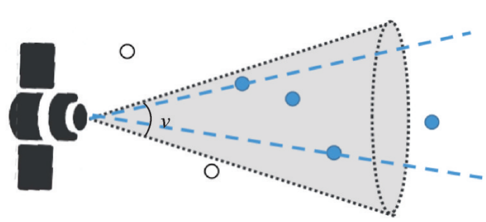

Fig.2 Field of view of a telescope (a circular field-of-view in this example)

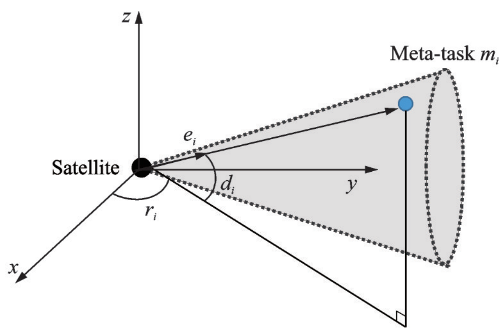

Fig.3 A meta-task in a 3D coordinate system

$$\cos\langle m_i, m_j \rangle = \boldsymbol{e}_i \cdot \boldsymbol{e}_j = \cos d_i \cos d_j \cos(r_i - r_j) + \sin d_i \sin d_j \tag{4}$$

where $i, j=n_1^k, n_2^k, \cdots, n_{|S_k|}^k$.

If a clustered task S_k satisfies this constraint, then $C_2^k=1$, otherwise $C_2^k=0$.

(2) Different clustered tasks should satisfy the following constraint: Observation times of the meta-task.

Meta-tasks are atomic. Each meta-task can only be observed once, i.e.,

$$\forall k, k': S_k \cap S_{k'} = \varnothing \tag{5}$$

where $k=1, 2, \cdots, N$; $k' \in C_{[1,N]}\{k\}$.

If a clustered task S_k satisfies this constraint, then $C_3^k=1$, otherwise $C_3^k=0$.

1.1.3 Multi-objective function

The multi-objective function f_1 that minimizes the number of clustered tasks and maximizes the constraint satisfaction degree is defined as

$$f_1 = \alpha_1 \cdot (1 - f_{\mathrm{cn}}) + \beta_1 \cdot f_{\mathrm{cs}} \tag{6}$$

where f_{cn} denotes the number of clustered tasks after normalization; f_{cs} the constraint satisfaction degree of the meta-tasks after normalization; and α_1 and β_1 are the factors of f_{cn} and f_{cs}, respectively. This paper sets $\alpha_1=1$ and $\beta_1=10$. A higher factor β_1 allows a task to satisfy each constraint as early as possible in the iteration. f_{cn} and f_{cs} are calculated as follows.

If there are fewer clustered tasks, the problem dimension in the scheduling will be smaller, resulting in more tasks being assigned.

$$f_{\mathrm{cn}} = \frac{N}{I} \tag{7}$$

The meta-tasks in each clustered task should satisfy clustering constraints as much as possible, and the normalized task constraint satisfaction degree is computed as

$$f_{\mathrm{cs}}=\frac{\sum_{k=1}^{N} C_1^k \cdot C_2^k \cdot C_3^k \cdot |S_k|}{I} \tag{8}$$

1.1.4 Recalculation for the properties of a clustered task

Meta-tasks are grouped into several clustered tasks by clustering, each containing at least one meta-task. The properties of a clustered task $S_k=\{m_i\}_{i=n_1^k}^{n_{|S_k|}^k}\in M'$ must be recalculated.

The priority of the clustered task is the average of all the meta-tasks in it, i.e.

$$p_k'=\frac{\sum_{i=n_1^k}^{n_{|S_k|}^k} p_i}{|S_k|} \tag{9}$$

The observation duration of the clustered task is the maximum among all its meta-tasks, i.e.

$$u_k'=\max(\{u_i\}_{i=n_1^k}^{n_{|S_k|}^k}) \tag{10}$$

1.2 Short-term task scheduling problem for astronomy satellites

Short-term scheduling typically generates weekly plans in minutes or seconds[1, 23]. The calculation precision in this paper is measured in minutes.

1.2.1 Definition and description of the symbols

A_{Sun}, A_{Earth} and A_{Moon} are the angles between the telescope pointing to and the directions of the the Sun, the Earth, and the Moon, respectively.

$G=\bigcup_{j=1}^{n_G} g_j$ denotes the set of periods when a satellite passes the ground station. $g_j=[g_j^{\mathrm{s}}, g_j^{\mathrm{e}}]\cap \mathbf{Z}$. g_j^{s} represents the start time of the jth period; g_j^{e} represents the end time of the jth period.

$H=\bigcup_{j=1}^{n_H} h_j$ is the set of periods when a satellite passes through the South Atlantic Anomaly (SAA). $h_j=[h_j^{\mathrm{s}}, h_j^{\mathrm{e}}]\cap \mathbf{Z}$, where h_j^{s} represents the start time of the jth period, and h_j^{e} the end time of the jth period.

$O_{i,j}$ is a decision variable, which determines the relationship between clustered tasks S_i and S_j in a schedule. If the next task of S_i is S_j, then $O_{i,j}=1$; otherwise, $O_{i,j}=0$.

t^{mane} is the attitude maneuver time of a satellite. The satellite must adjust its attitude before each observation. In order to simplify the calculation, the attitude maneuver time is taken as the maximum value, i.e., $t^{\mathrm{mane}}=5$.

$T=[t_1, t_2, \cdots, t_N]$ denotes the observation start time of the clustered tasks, in which t_i is the start time of S_i.

1.2.2 Constraints for scheduling

(1) Pointing of the payload

The observation will be impacted by sunlight, Earth's occlusion, and lunar albedo. Consequently, the payload's pointing must satisfy the following requirements

$$\begin{cases} A_{\mathrm{Sun}}\geqslant \alpha \\ A_{\mathrm{Moon}}\geqslant \beta \\ A_{\mathrm{Earth}}\geqslant \gamma \end{cases} \tag{11}$$

The values of A_{Sun}, A_{Moon} and A_{Moon} are calculated on the Satelite Tool Kit software. Referring to the given existing tasks, we set $\alpha=95°$, $\beta=20°$, and $\gamma=77°$.

If a clustered task S_k satisfies this constraint, then $C_4^k=1$; otherwise $C_4^k=0$.

(2) No attitude maneuvering when a satellite passes the ground stations

The satellite will downlink data as it passes the ground station. It should not perform attitude maneuver μ minutes before and during the transit, i.e.

$$\forall i:[t_i-t^{\mathrm{mane}}, t_i]\cap \bigcup_{j=1}^{n_G}[g_j^{\mathrm{s}}-\mu, g_j^{\mathrm{e}}]=\varnothing \tag{12}$$

where $i=1, 2, \cdots, N$.

According to the performance of an astronomical satellite, μ is set to 6 empirically in the following simulation.

If a clustered task S_k satisfies this constraint, then $C_5^k=1$, otherwise $C_5^k=0$.

(3) No observation when a satellite passes the SAA

When passing SAA, the satellite deactivates the payload to prevent radiation damage, i.e.

$$\forall i:[t_i, t_i+u_i]\cap H=\varnothing \tag{13}$$

where $i=1, 2, \cdots, N$.

If a clustered task S_k satisfies this constraint, then $C_6^k=1$, otherwise $C_6^k=0$.

(4) Maximum number of attitude maneuvers per orbit

Due to limited energy, the satellite should perform at most X times attitude maneuvers in an orbit period, i.e.

$$\forall \underbrace{i,k,\cdots,l,j}_{X}=1,2,\cdots,N:$$
$$(t_j-t_i\geqslant b)\wedge(\underbrace{O_{i,k}\cdots O_{l,j}}_{X-1}=1) \quad (14)$$

where b is the orbit period of the satellite.

We set $X=3$[24] in the simulation. If a clustered task S_k satisfies this constraint, then $C_7^k=1$, otherwise $C_7^k=0$.

(5) Satellite observation capability

A satellite can only carry out one observation task at a given time, i.e.

$$\forall i,j:[t_i,t_i+u_i]\cap[t_j,t_j+u_j]=\varnothing \quad (15)$$

where $i,j=1,2,\cdots,N$, and $j\neq i$.

If a clustered task S_k satisfies this constraint, then $C_8^k=1$, otherwise $C_8^k=0$.

(6) Time of attitude maneuver

Before executing their next task, satellites need time to perform attitude maneuvers, i.e.

$$\forall i,j=1,2,\cdots,N:$$
$$(t_i+u_i+t^{\text{mane}}\leqslant t_j)\wedge(O_{i,j}=1) \quad (16)$$

If a clustered task S_k satisfies this constraint, then $C_9^k=1$, otherwise $C_9^k=0$.

1.2.3 Multi-objective function

Short-term task scheduling aims to include as many tasks in the schedule as feasible. It furthermore endeavors to raise the average priority of the scheduled tasks to the highest possible level. The multi-objective function is designed as

$$f_2=\alpha_2\cdot f_{\text{pri}}+\beta_2\cdot f_{\text{TC}} \quad (17)$$

where f_{pri} is the normalized priority of clustered tasks; f_{TC} the meta-tasks' completion; and α_2 and β_2 are the factors of f_{pri} and f_{TC}, respectively. They have the relationship $\alpha_2+\beta_2=1$. In this paper, we set $\alpha_2=\beta_2=0.5$.

A task with a high priority is scientifically important. The normalized task priority is determined according to the following equation

$$f_{\text{pri}}=\frac{\sum_{k=1}^{N}\lambda_k\cdot p'_k}{\sum_{k=1}^{N}p'_k} \quad (18)$$

where $\lambda_k=\prod_{i=4}^{9}C_i^k,\ k=1,2,\cdots,N$.

More tasks scheduled result in a higher level of task completion rate and a greater likelihood of scientific discovery. Therefore, the normalized task completion is calculated as

$$f_{\text{TC}}=\frac{\sum_{k=1}^{N}\lambda_k}{N} \quad (19)$$

where $\lambda_k=\prod_{i=4}^{9}C_i^k,\ k=1,2,\cdots,N$.

2 Modified ABC Algorithm

The ABC algorithm[25] is an intelligent optimization algorithm inspired by the cooperative behavior of bee colonies when foraging for food. This algorithm searches for the global optimal solution via the localized optimization of individual bees. Compared with other intelligent algorithms, ABC has the advantages of fewer control parameters and stronger optimization ability. Therefore, this algorithm has been widely used and applied in search tasks such as path planning[26-28], disease identification[29-30], parameter tuning of controllers[31], and car structure design[32], etc.

2.1 BasicABC algorithm

Bees are highly social creatures. They play three roles: Employed bees, onlooker bees, and scout bees. As detailed in Table 1, each of these roles has distinct duties in the food collection process. The collective goal of the bee population is to find the best food source.

Table 1 Performance comparison between the conventional microwave system and the microwave photonic system

Role	Division of labor in the bee population	Functions in the algorithm
Employed bee	Search for food sources and share the information with other bees	Explore excellent potential solutions
Onlooker bee	Receive information from employed bees; pick an employed bee and follow it to search for food nearby	Utilize the information to search for better solutions
Scout bee	Transformed from employed bees with depleted food sources and select food sources randomly	Jump out of the local optimal solution

2.1.1 Population initialization

The number of the population is N_P. Each food source is a solution. These two terms will be used interchangeably. At first, there are only employed bees and onlooker bees. Their numbers each comprise half of the population size and are noted as N_S. The number of food sources is equal to the number of employed bees.

For an L-dimensional problem, the position of a food source is denoted as $X_i=[x_{i,1}, x_{i,2}, \cdots, x_{i,L}]$, where $i=1, 2, \cdots, N_S$. Each dimension of the initial food source is generated by

$$x_{i,j}=L_j+\text{rand}\cdot(U_j-L_j) \tag{20}$$

where L_j and U_j are the lower bound and the upper bound of the dimension, respectively; rand is a function generating a random number within the range $[0,1]$.

2.1.2 Employed bee phase

Every employed bee randomly selects a dimension j of its food source to update according to Eq.(21). The updated new food source is noted as V_i.

$$v_{i,j}=x_{i,j}+\varphi_{i,j}(x_{i,j}-x_{k,j}) \tag{21}$$

where $\varphi_{i,j}$ is a random number within the range $[-1, 1]$. X_k is a food source selected randomly, where $k=1, 2, \cdots, N_S$, $k\neq i$.

Then, the employed bee chooses the better one between V_i and X_i based on a greedy strategy. If the food source is updated, it makes the search count $\text{trial}_i=0$ for food source X_i; otherwise, $\text{trial}_i=\text{trial}_i+1$.

2.1.3 Onlooker bee phase

When the employed bees return to their nest, they share information about food sources with onlooker bees. Each onlooker bee chooses a food source to search for according to the probability calculated by

$$p_i=\frac{f_i}{f_{\max}} \tag{22}$$

where p_i denotes the probability of food source X_i being chosen. f_i the fitness value of food source X_i, $f_{\max}=\max(f_1, f_2, \cdots, f_{N_S})$. After selecting an employed bee, an onlooker bee searches in the neighborhood of its food source. The onlooker bee uses the same search strategy as the employed bees by Eq.(21).

Then, the onlooker bee chooses the better one between V_i and X_i based on a greedy strategy. If the food source is updated, then it makes the search count $\text{trial}_i=0$ for food source X_i; otherwise, $\text{trial}_i=\text{trial}_i+1$.

2.1.4 Scout bee phase

At the end of each iteration, check if any food sources have been searched more than limilted times. If so, the related employed bee is converted to a scout bee and randomly generates a new food source by Eq.(20).

2.2 DirectABC algorithm

A new solution in the basicABC algorithm is generated based on an old solution. Such a simple operation is suitable for local search, but it leads to the inability of good information to spread within the population quickly. Therefore, ABC has a limited global search capability and a slow convergence speed[33].

As was previously stated, onlooker bees obtain information about the food source from the employed bees. This distinction in search strategies between employed and onlooker bees is not considered in basicABC. Inspired by the elite search strategy[34] and the quick ABC algorithm[35], the ABC algorithm with adaptive hybrid search strategies (DirectABC) is proposed.

2.2.1 Employed bee: Adaptive elite global-local search strategy

In the elite-guided search strategy, a bee randomly selects a solution in the population as the search center and then chooses a random elite solution to guide the search. This strategy not only enlarges the search space but also improves the ability to explore optimal solutions and accelerates the convergence speed of the algorithm. The equation of the search process is given as

$$v_{i,j}=x_{k,j}+\varphi_{i,j}(x_{l,j}^{\text{elite}}-x_{k,j}) \tag{23}$$

where $x_{k,j}$ is the jth dimension of solution X_k, and the random search center, $k=1, 2, \cdots, N_S$. x_l^{elite} is the lth elite solution, where $l=1, 2, \cdots, N_e$. $N_e=\lfloor\eta\cdot N_S\rfloor$, where η denotes the elite rate and $\lfloor\cdot\rfloor$ a round down function.

This strategy accelerates the algorithm's convergence, but it introduces the premature convergence problem, which implies that all individuals in the population are trapped in local optima. The ABC algorithm is expected to perform global exploration in the early iterations and more local exploration in the later iterations. An adaptive elite global-local search strategy is proposed to improve the solution accuracy and overcome the premature convergence issue. The improved strategy works as

$$v_{i,j}=\delta_{i,j}^{t}\left[x_{k,j}+\varphi_{i,j}\left(x_{l,j}^{\text{elite}}-x_{k,j}\right)\right]+(1-\delta_{i,j}^{t})\left[x_{i,j}+\varphi_{i,j}(x_{i,j}-x_{k,j})\right] \tag{24}$$

where $\delta_{i,j}^{t}$ is a strategy control factor. If rand $\geqslant\varepsilon$, $\delta_{i,j}^{t}=1$; otherwise, $\delta_{i,j}^{t}=0$. $\varepsilon=0.5\times(1+\cos((t\cdot\pi)/C_{\max}))$, where t is the current iteration; and $C_{\max}$ the maximum iteration.

2.2.2 Onlooker bee: Adaptive variable neighborhood optimal search strategy

Quick ABC (qABC) realizes the difference between onlooker bees and employed bees to improve the way onlooker bees search. The best solution in the neighborhood of the current solution is taken as the search center. Subsequently, updates are guided by a randomly selected solution by

$$v_{B_i,j}^{\text{best}}=x_{B_i,j}^{\text{best}}+\varphi_{i,j}(x_{B_i,j}^{\text{best}}-x_{k,j}) \tag{25}$$

where B_i represents the neighbors of X_i and itself; and $x_{B_i}^{\text{best}}$ the best solution among B_i.

The definition of B_i is given as

$$B_i=\{l|d(i,l)\leqslant\sigma\cdot d_i^{\text{mean}}\} \tag{26}$$

where $d(i,l)$ calculates the Euclidean distance from X_i to X_l, and $l=1,2,\cdots,N_S$. d_i^{mean} is the average Euclidean distance between X_i and the rest of solutions, i.e.

$$d_i^{\text{mean}}=\frac{\sum_{l=1}^{N_S}d(i,l)}{N_S-1} \tag{27}$$

σ is the neighborhood radius, which controls the neighborhood size; $\sigma\in[0,+\infty)$. When $\sigma=0$, $x_{B_i,j}^{\text{best}}=x_{i,j}$, Eq.(25) is equivalent to Eq.(21). When $\sigma\rightarrow+\infty$, Eq.(25) updates solutions with the global optimum as the search center. However, the performance of qABC is affected by the parameters σ. It is anticipated that the algorithm should prioritize global exploration in the early iterations and local exploration in the later iterations. To this end, an adaptive variable neighborhood optimal search strategy is proposed by introducing an adaptive operator. The modified strategy searches by Eq.(25). But the operator σ_t and the neighbors are calculated by Eq.(28) and Eq.(29), respectively.

$$\sigma_t=\left(\frac{t}{C_{\max}}\right)^{-\frac{3}{2}}-1 \tag{28}$$

$$B_i=\{l|d(i,l)\leqslant\sigma_t\cdot d_i^{\text{mean}}\} \tag{29}$$

In Eq.(28), $C_{\max}$ is the maximum iteration, and $t\in\{1,2,\cdots,C_{\max}\}$ denotes the current iteration. The operator σ_t approaches infinity at the start of the iteration, then decreases monotonically with iteration and eventually converges to 0. Consequently, this method enables a transition from global search to local search.

2.2.3 Pseudocode of DirectABC

The inputs of DirectABC are as follows: The number of population N_P, the elite rate η, the maximum iteration $C_{\max}$, and the maximum search times limit. The detailed steps of DirectABC are provided as follows.

(1) //Initialization phase:

(2) Initialize the food sources(initial solutions) X_i by Eq. (20), where $i=1,2,\cdots,N_S$.

(3) Calculate the fitness of solutions.

(4) Memorize the best solution

(5) $t=0$

(6) Repeat

(7) //Employed bee phase:

(8) For each employed bee:

(9) Search a new candidate solution V_i according to Eq.(24) and calculate its fitness.

(10) Apply the greedy strategy between V_i and X_i.

(11) If there is an update, trial$_i=0$; otherwise, trial$_i=$ trial$_i+1$.

(12) //Onlooker bee phase:

(13) For each onlooker bee:

(14) Select a solution X_i according to Eq.(22).

(15) Determine the best neighbor of X_i depending on Eqs. (25,28,29), and calculate its fitness.

(16) Apply the greedy strategy between $V_{B_i}^{\text{best}}$ and $X_{B_i}^{\text{best}}$.

(17) If there is an update, $\text{trial}_{B_i}^{\text{best}}=0$; otherwise, $\text{trial}_{B_i}^{\text{best}}=\text{trial}_{B_i}^{\text{best}}+1$.

(18) Memorize the best solution found so far.

(19) //Scout bee phase:

(20) Check whether any solution has been searched for more than limited times. If it exists, abandon the solution, and replace it with a new one generated by Eq.(20).

(21) $t=t+1$

(22) Until $t=C_{\max}$

The flowchart of DirectABC is provided in Fig.4.

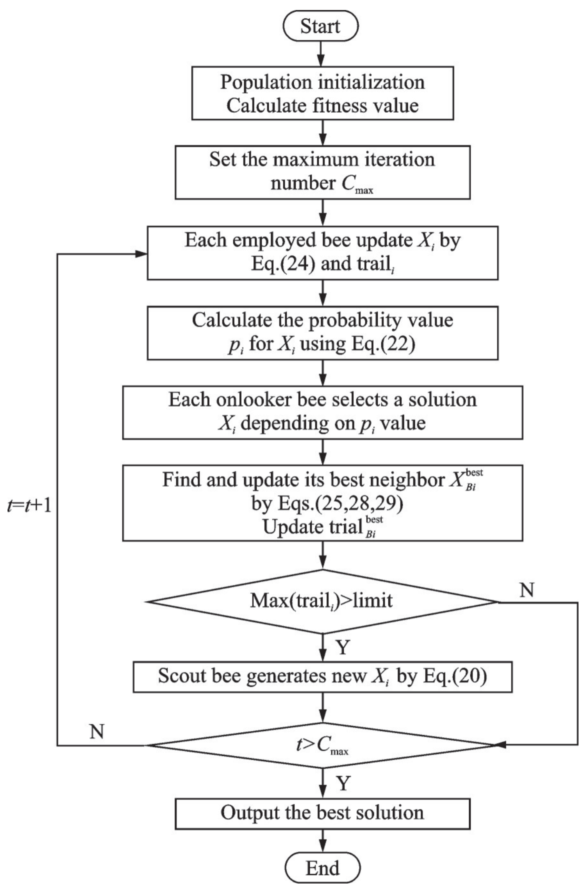

Fig.4 Flowchart of DirectABC

3 Simulation and Result Analysis

3.1 Data details

This experiment uses the simulation dataset of a satellite. The dataset contains long-term scheduling results for 52 weeks (one year), with a time precision of days. To verify the algorithm's and framework's effectiveness, we selected weekly long-term scheduling results of different dimensions for simulation. Table 2 shows the dataset-related information.

Table 2 Statistics of datasets

Dataset	Week	Number of meta-tasks	Number of tasks with priority of 1	Number of tasks with priority of 2	Number of tasks with priority of 3	Number of tasks with priority of 4
Dataset 1	4	104	1	20	61	22
Dataset 2	6	201	61	67	50	23
Dataset 3	13	310	79	73	128	30
Dataset 4	31	411	314	42	24	31
Dataset 5	10	509	360	65	62	22
Dataset 6	51	592	508	49	21	14

3.2 Solution structure

Fig.5(a) shows a clustering solution. The clustering algorithm assigns a label l_i to each meta-task m_i, which records the category that the meta-task belongs, where $l_i\in[1,I]$, $i=1,2,\cdots,I$. Meta-tasks with the same label consist of a clustered task. There are N different labels in total, which means it has N clustered tasks, where $N\in[1,I]$. Fig.5(b) demonstrates a solution for scheduling. The scheduling algorithm allocates a start time t_j for each clustered task S_j, where $t_j\in[1, 10\,080]$ and $j=1,2,\cdots,N$.

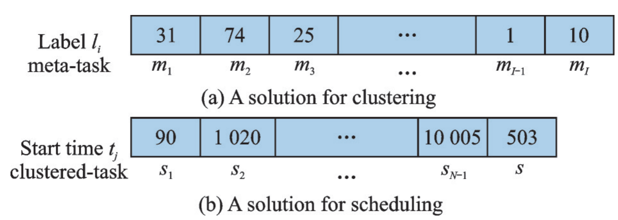

Fig.5 Forms of the algorithm solution

3.3 Experimental setting

As shown in Table 3, six algorithm combinations are experimentally evaluated. For each combination, the clustering algorithm runs five times and the scheduling algorithm runs ten times. Besides, two widely-used algorithms, the genetic algorithm

(GA) and the particle swarm optimization (PSO), are also used for comparison in the clustering phase.

Table 3 Algorithm combinations for experiments

Case	Algorithm	
	Clustering	Scheduling
1	DirectABC	DirectABC
2		BasicABC
3	BasicABC	DirectABC
4		BasicABC
5	—	DirectABC
6		BasicABC

The common parameters are set as follows: $N_P = 200$; limit $= 100$; C_{max} is set to 1 000 in clustering and 10 000 in scheduling. In DirectABC, the elite rate $\eta = 0.1$. In standard PSO, inertia weight $w = 1$, local acceleration factor $c_1 = 2$, and global acceleration factor $c_2 = 2$[36]. Standard GA has two additional parameters: The probability of crossover p_c and the probability of mutation p_m. According to Pongcharoen's setting in GA[37], experiments are performed for three parameter combinations: $p_c=0.3$ and $p_m=0.02$, $p_c=0.6$ and $p_m=0.1$, and $p_c=0.9$ and $p_m=0.18$. The field of view of the payload $v=15°$.

3.4 Performance index

N: The number of clustered tasks. Each clustered task contains several meta-tasks. The number of meta-tasks in all clustered tasks equals the number of meta-tasks before clustering.

τ(%): Task number reduction efficiency. The number of clustered tasks is lower with a higher τ.

$$\tau = 1 - f_{cn} \tag{30}$$

T_C(%): Task completion rate. The ratio of the number of meta-tasks in the scheduling result to the one in that dataset.

$$T_C = \frac{\sum_{k=1}^{N} \lambda_k \cdot |S_k|}{I} \tag{31}$$

$T_{C(p=k)}$(%): Task completion rate of meta-tasks with priority k.

ACS: Average clustering constraint satisfaction. It demonstrates the ability of an algorithm to find solutions that satisfy the constraints. It is calculated by

$$\mathrm{ACS} = \frac{C_1^k \cdot C_2^k \cdot C_3^k}{T_{run}} \tag{32}$$

where T_{run} is the number of times that the clustering experiment runs.

Speed of convergence: If the algorithm converges quickly, it should obtain relatively stable fitness in a relatively small number of iterations.

3.5 Results and analysis

Scheduling arranges the clustered tasks onto a timeline measured in minutes. The scheduling algorithm outputs the start time of each clustered task and task-related information. Due to the space limitation, only a subset of the optimal experiment results for Case 1 on Dataset 5 are shown in Table 4.

3.5.1 Effectiveness of clustering

The effectiveness of clustering is evaluated from two perspectives: Task number reduction efficiency and task completion rate.

Table 4 Optimal partial scheduling results obtained by applying Case 1 on Dataset 5

Observing order	Set of target number	ra/(°)	dec/(°)	Start time/min	End time/min
1	[159,175,253]	[208.82,215.66,233.75]	[56.21,58.03,53.34]	111	131
2	[119]	[169.28]	[20.24]	155	175
⋮	⋮	⋮	⋮	⋮	⋮
99	[492,347,505,384,444]	[208.82,215.66,199.99,224.01,215.37]	[56.21,58.03,52.59,50.80,47.79]	4 323	4 343
100	[455,445]	[213.41,185.44]	[70.50,75.31]	4 378	4 398
⋮	⋮	⋮	⋮	⋮	⋮
237	[492,505,444,593]	[208.81,199.99,215.37,214.25]	[56.21,52.59,47.79,44.93]	9 957	9 977
238	[557,410]	[169.29,148.89]	[65.37,69.06]	9 982	10 002
239	[169,314,467]	[220.53,207.22,208.58]	[35.44,26.59,32.93]	10 037	10 057

(1) Task number reduction efficiency

Fig.6 compares the average number of tasks before and after clustering with different algorithms.

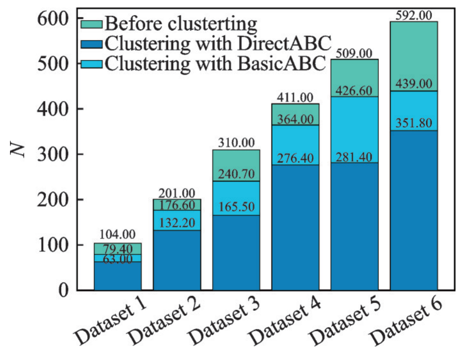

Fig.6 Comparison of the average number of tasks before and after clustering with different algorithms

DirectABC clustering reduced the average number of clustered tasks on Datasets 1—6 to 63, 132.2, 165.5, 276.4, 281.4, and 351.8, respectively. The average task number reduction efficiency $\tau=$ 39.72%. On Datasets 1—6, the BasicABC algorithm generated an average of 79.4, 176.6, 165.5, 276.4, 281.4, and 351.8 clustered tasks, respectively. The average task number reduction efficiency $\tau=$ 18.60%.

Therefore, regardless of whether DirectABC or BasicABC is applied, a significant decrease in task numbers can be observed, proving the effectiveness of the clustering model.

(2) Task completion rate

Fig.7 displays the task completion rate for different algorithm combinations.

When DirectABC is used for scheduling, the task completion rates with clustering (Cases 1 and 3) are higher than those without clustering (Case 5).

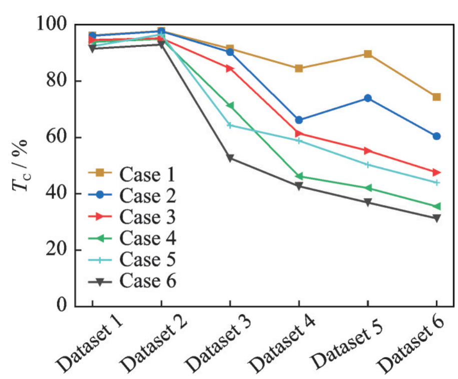

Fig.7 T_c of different algorithm combina-tions on different datasets

Similarly, when BasicABC is used for scheduling, the task completion rates with clustering (Cases 2 and 4) are also higher than those without clustering (Case 6) in general.

Clustering can therefore effectively improve task completion rates in scheduling.

3.5.2 Effectiveness of DirectABC

The effectiveness of the DirectABC algorithm will be evaluated based on the following factors: Task number reduction efficiency, task completion rate, task completions with different priorities, and convergence speed. In addition, PSO and GA will be applied in the clustering phase to validate DirectABC's performance.

(1) Task number reduction efficiency

As discussed in Section 3.5.1, the τ-value of clustering with BasicABC is 18.60%, while that with DirectABC is 39.72%. As a result, the DirectABC algorithm performs more effectively in the clustering phase.

(2) Task completion rate

Now compare DirectABC and BasicABC in the scheduling phase based on the same clustering results, as shown in Fig.7.

When using DirectABC in the clustering phase, the task completion rates with DirectABC scheduling are 0.17%, 0.09%, 1.25%, 18.32%, 15.71%, and 13.93% higher than those with BasicABC on Datasets 1—6, respectively.

When using BasicABC in the clustering phase, the task completion rates with DirectABC scheduling are 0.66%, 0.18%, 13.25%, 15.15%, 13.18%, and 11.99% higher than those with BasicABC on Datasets 1—6, respectively.

When no clustering is used, the task completion rates with DirectABC scheduling are 0.81%, 1.57%, 11.6%, 15.8%, 13.36%, and 12.65% higher than those with BasicABC on Datasets 1—6, respectively.

Therefore, under the same clustering algorithm, DirectABC consistently achieves a higher task completion rate in the scheduling phase than BasicABC.

(3) Task completions with different priorities

Fig.8 shows the average task completion rates of meta-tasks with different priorities. Except for Dataset 2 at p=4, where Case 1 had a marginally lower completion rate than that of Case 2, Case 1 had the best completion rate for both high and low-priority tasks. In addition, Case 1 has a lower standard deviation, indicating that DirectABC performs greater stably.

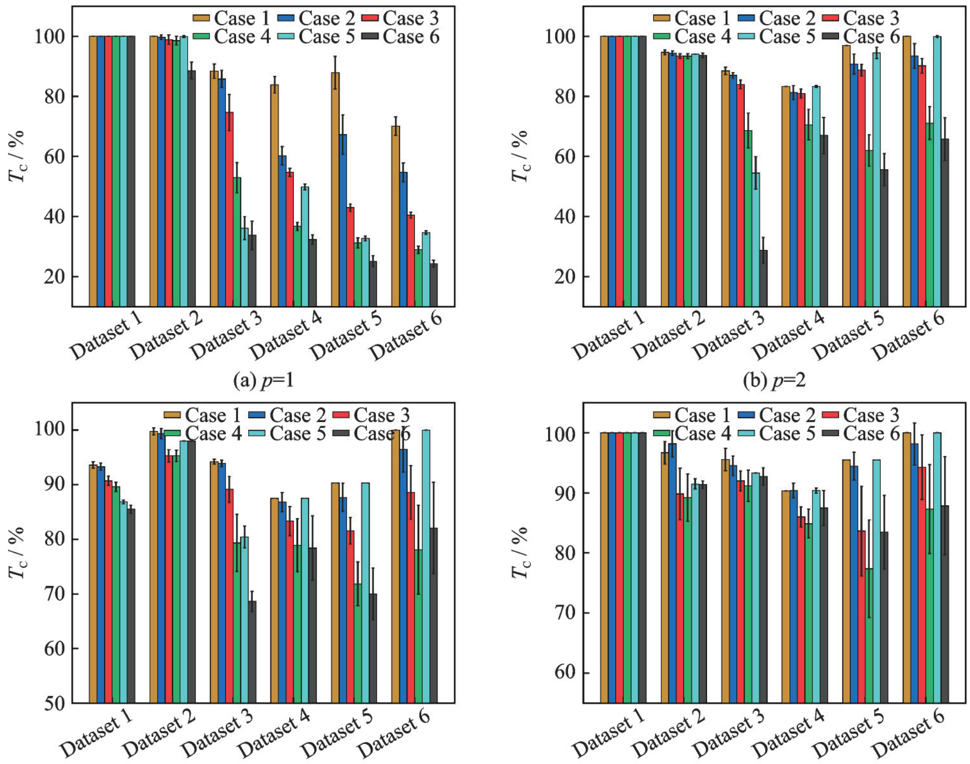

Fig.8 Average task completion by priority for different combinations of algorithms on different datasets

(4) Convergence speed

Fig.9 demonstrates that both PSO and GA have difficulty in finding feasible solutions that fully satisfy the clustering constraints in 1 000 iterations. In contrast, DirectABC achieves superior clustering results than GA and PSO, finding solutions that satisfy all constraints more quickly.

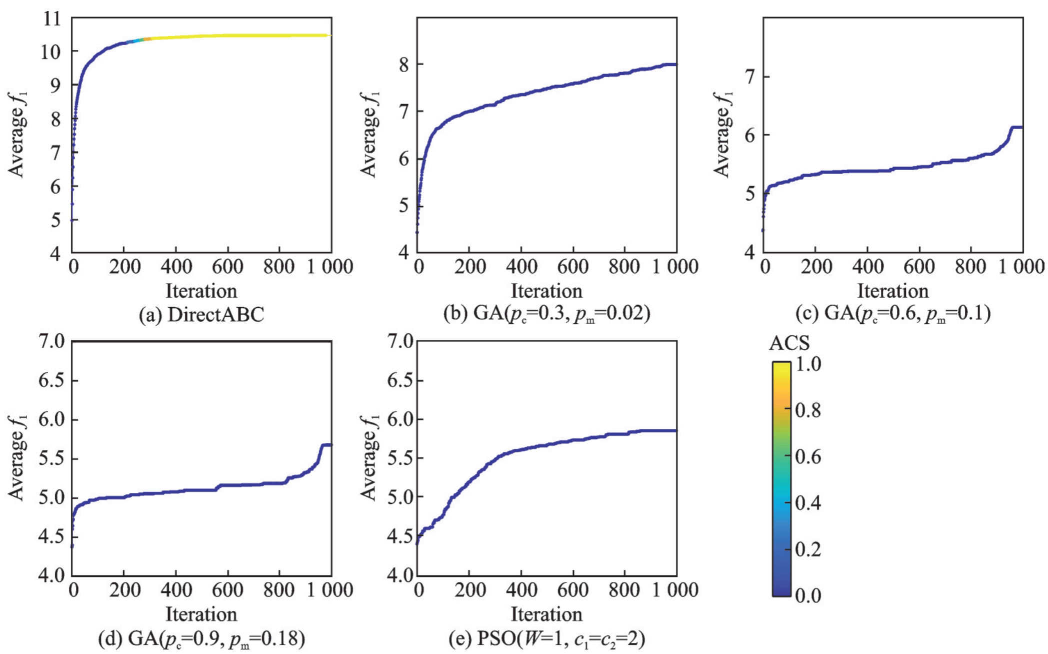

Fig.9 Average convergence curves of different algorithms for clustering on Dataset 5

As shown in Fig.10, DirectABC also converges more quickly during the scheduling phase. Case 1, Case 3, and Case 5 always converge earlier with higher accuracy than their experimental counterparts, Case 2, Case 4, and Case 6, respectively, on different datasets.

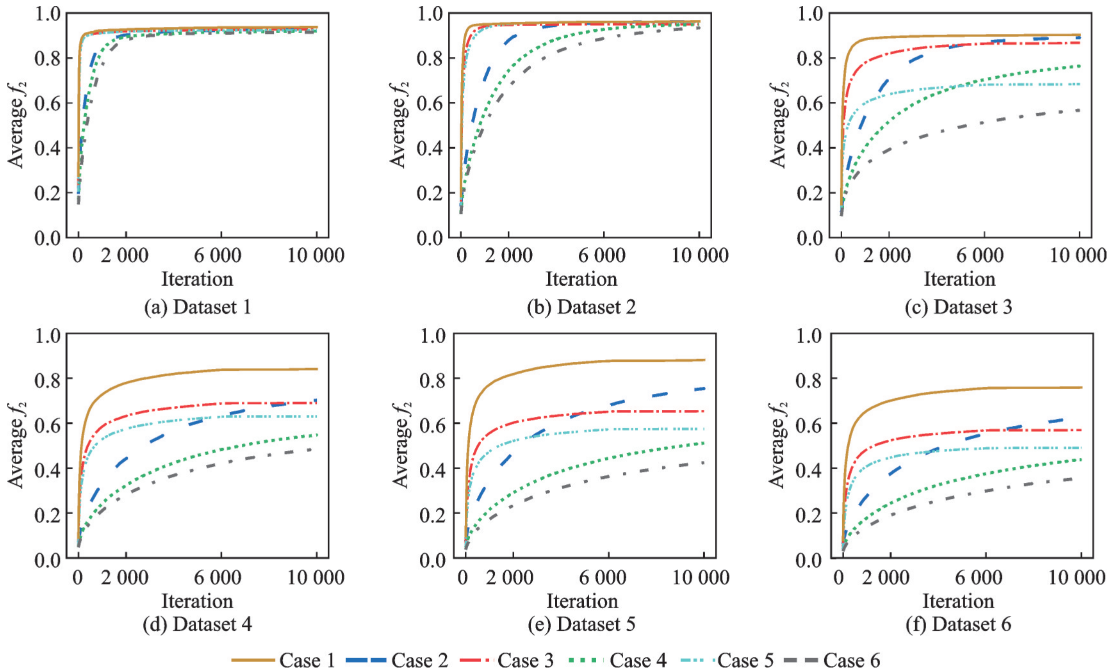

Fig.10 Average convergence curves of different algorithm combinations for scheduling on different datasets

4 Discussion and Conclusions

This paper proposes a clustering-scheduling framework suitable for astronomy satellite task scheduling in the context of oversubscribed tasks. In the clustering phase, a multi-objective clustering model was established considering the constraints of the observation time window, payload's field-of-view, and task observation times. The purpose of clustering is to reduce the task scale by grouping several meta-tasks into a single clustered task, which is then observed simultaneously by the payload. In the scheduling phase, we consider six constraints about the space environment, user requirements, and satellite resources and develop a multi-objective scheduling model. The model aims to maximize the task completion rate and the priority of assigned tasks.

The ABC algorithm has few control parameters. It is particularly effective at solving high-dimensional problems. These characteristics make it an ideal candidate for the clustering-scheduling problem. However, BasicABC has a slow convergence speed and inadequate global search capabilities, which prompts the proposal of DirectABC. DirectABC incorporates the adaptive elite global-local search strategy into the employed bee phase. This strategy causes the algorithm to focus on global exploration early on and mainly conduct local searching later on. In addition, the onlooker bee phase applies an adaptive variable neighborhood optimal search strategy. This strategy takes the optimal solution in the neighborhood as the search center, with a random solution guiding the updates. The choice of neighborhood factor is crucial to the performance of the algorithm. Therefore, the δ_t operator is designed to decrease monotonically with iteration, enabling an adaptive adjustment of the neighborhood from large to small.

The following conclusions are drawn from simulations of datasets in six dimensions:

(1) The framework proposed in this paper is suitable for the oversubscribed task scheduling problem.

① The size of the tasks can be effectively reduced during the clustering phase. For instance, the

DirectABC algorithm reduced 592 meta-tasks to 351.8 clustered tasks on Dataset 6, with an average task number reduction efficiency of 40.57%.

② Clustering improves task completion rate in scheduling. For example, Case 1 and Case 5 both used the same scheduling algorithm. However, Case 1 with DirectABC clustering improved the task completion rates by 3.94%, 1.25%, 27.23%, 25.68%, 39.31%, and 30.38%, respectively, on each dataset compared to Case 5 without clustering.

(2) The DirectABC algorithm has superior performance over BasicABC, PSO and GA.

① DirectABC obtains a better solution. Regarding clustering, the average task number reduction efficiency of DirectABC was 21.12% higher than that of BasicABC. In addition, DirectABC could find a solution that satisfies the constraints quickly. while GA and PSO were prone to fall into local optimal solutions leading to low efficiency. In terms of scheduling, taking Case 1 and Case 2 as examples, they both used the same clustering algorithm. Regarding average task completion rate, Case 1 using DirectABC scheduling was 0.17%, 0.09%, 1.25%, 18.32%, 15.71%, and 13.93% higher than Case 2 using BasicABC scheduling on each dataset, respectively.

② DirectABC performs more stably. Case 1, which used DirectABC clustering and scheduling, had a lower standard deviation than other cases in general.

③ DirectABC has a faster convergence speed. It can find a superior solution earlier. In the clustering phase of Dataset 5, DirectABC found feasible solutions after 300 iterations, while GA and PSO failed to show a significant convergence trend within 1 000 iterations. In the scheduling phase, cases with DirectABC scheduling converged faster than cases with BasicABC based on the same clustering method.

The framework still needs to be improved. And clustering is only applied before short-term task scheduling. Whether clustering can be carried out before long-term scheduling is also a direction for future exploration. Plus, all of the datasets used in this paper are deterministic tasks. When transient targets like gamma-ray bursts appear, users propose the observation requirements for these uncertainty tasks. These tasks usually have a higher priority and need to be observed immediately. Therefore, the framework should also consider dynamic rescheduling of responding quickly to uncertain and interrupted tasks.

References

[1] LI Dalin. Research on observation scheduling model and method of astronomy satellite[D]. Harbin: Harbin Institute of Technology, 2021. (in Chinese)

[2] SMITH D E. Choosing objectives in over subscription planning[C]//Proceedings of the 14th International Conference on Automated Planning and Scheduling. Whistler, Canada: AAAI, 2003: 393-401.

[3] JOHNSTON M D, MILLER G. Spike: Intelligent scheduling of hubble space telescope observations[J]. Intelligent Scheduling, 1994, 19: 3-5.

[4] TOHUVAVOHU A. Improving science yield for NASA swift with automated planning technologies[C]//Proceedings of the 18th International Workshop on Advanced Computing and Analysis Techniques in Physics Research (ACAT 2017). Seattle, USA: IOP, 2018: 032010.

[5] CASTELLINI F, LAVAGNA M. Advanced planning and scheduling initiative XMAS tool: AI for automatic scheduling of XMM-Newton long term plans[C]//Proceedings of the 6th International Workshop on Planning and Scheduling for Space. Pasadena, USA: JPL, 2009.

[6] CEDRIC P, GERARD V. AIMS: A tool for long-term planning of the ESA INTEGRAL mission[C]//Proceedings of the 6th International Workshop on Planning and Scheduling for Space. Pasadena, USA: JPL, 2009.

[7] CESTA A, CORTELLESSA G, FRATINI S, et al. MrSPOCK: A long-term planning tool for MARS EXPRESS[C]//Proceedings of the 6th International Workshop on Planning and Scheduling for Space. Pasadena, USA: JPL, 2009.

[8] PRALET C, ROUSSEL S, JAUBERT J, et al. Planning for space telescopes: Survey, case studies, and lessons learned[J]. Aerospace Lab, 2020, 15(5): 1-17.

[9] MORALES J C, BEAULIEU J P, FORESTO V C, et al. Scheduling the EChO survey with known exoplanets[J]. Experimental Astronomy, 2015, 40: 655-670.

[10] HUANG Y, QU J L, JIA S M, et al. Long-term

planning algorithm for the HXMT mission[J]. Chinese Journal of Space Science, 2017, 37(6): 766-772.

[11] JAUBERT J, DEBOUT V. Mission planning and scheduling activities conducted at French Space Agency (CNES) for the SVOM mission[C]//Proceedings of the 11th International Workshop on Planning and Scheduling for Space. Berkeley, USA: JPL, 2019.

[12] LI D L, GU Y F, JAUBERT J, et al. A robust scheduling algorithm for space telescopes with unpredictable tasks[J]. Science China Technological Sciences, 2021, 64(3): 571-584.

[13] WU H Y, MENG X, ZHANG Y Z, et al. Research on the planning method for astronomy observation mission[J]. Chinese Journal of Space Science, 2013, 33(5): 561-568.

[14] LIU W, LI L G. Mission planning of space astronomical satellite based on improved genetic algorithm[J]. Computer Simulation, 2014, 31(12): 54-58.

[15] SHE Y, LI S. Optimal slew path planning for the Sino-French Space-based multiband astronomical Variable Objects Monitor mission[J]. Journal of Astronomical Telescopes, Instruments, and Systems, 2018, 4(1): 017001.

[16] XU Ziling. Research on Astronomical Satellite ToO planning under Tiling coverage strategy[D]. Beijing: University of Chinese Academy of Sciences, 2021. (in Chinese)

[17] XU Z L, LIU Y R, FENG Z. Mission planning for astronomical satellite based on genetic algorithm under tiling coverage strategy[J]. Chinese Journal of Space Science, 2022, 42(2): 321-328.

[18] ZHAO Y, DU B, LI S. Agile satellite mission planning via task clustering and double-layer Tabu algorithm[J]. Computer Modeling in Engineering & Sciences, 2020, 122(1): 235-257.

[19] DU B, LI S. A new multi-satellite autonomous mission allocation and planning method[J]. Acta Astronautica, 2019, 163: 287-298.

[20] SHE Y C, LI S, ZHAO Y B. Onboard mission planning for agile satellite using modified mixed-integer linear programming[J]. Aerospace Science and Technology, 2018, 72: 204-216.

[21] DU B, LI S, SHE Y C, et al. Area targets observation mission planning of agile satellite considering the drift angle constraint[J]. Journal of Astronomical Telescopes, Instruments, and Systems, 2018, 4(4): 1-19.

[22] SAXENA A, PRASAD M, GUPTA A, et al. A review of clustering techniques and developments[J]. Neurocomputing, 2017, 267: 664-681.

[23] WU J. Space science introduction[M]. Beijing: Science Press, 2020.

[24] LI L, LIU F, FENG H, et al. Einstein probe: Exploring the ever-changing X-ray universe[J]. Scientia Sinica: Physica, Mechanica & Astronomica, 2018, 48(3): 039502.

[25] KARABOGA D. An idea based on honey bee swarm for numerical optimization: Technical report-TR06[R]. Kayseri, Turkey: [s.n.], 2005.

[26] CONTRERAS-CRUZ M A, AYALA-RAMIREZ V, HERNANDEZ-BELMONTE U H. Mobile robot path planning using artificial bee colony and evolutionary programming[J]. Applied Soft Computing, 2015, 30: 319-328.

[27] HAN Z, CHEN M, SHAP S, et al. Improved artificial bee colony algorithm-based path planning of unmanned autonomous helicopter using multi-strategy evolutionary learning[J]. Aerospace Science and Technology, 2022, 122: 107374.

[28] XU F, LI H, PUN C M, et al. A new global best guided artificial bee colony algorithm with application in robot path planning[J]. Applied Soft Computing, 2020, 88: 106037.

[29] SUBANYA B, RAJALAXMI R. Feature selection using artificial bee colony for cardiovascular disease classification[C]//Proceedings of 2014 International Conference on Electronics and Communication Systems (ICECS). [S.l.]: IEEE, 2014: 1-6.

[30] COLETO-ALCUDIA V, VEGA-RODRÍGUEZ M A. Artificial bee colony algorithm based on dominance (ABCD) for a hybrid gene selection method[J]. Knowledge-Based Systems, 2020, 205: 106323.

[31] MISHRA A K, TIWARI V K, KUMAR R, et al. Speed control of DC motor using artificial bee colony optimization technique[C]//Proceedings of 2013 International Conference on Control, Automation, Robotics and Embedded Systems (CARE). [S. l.]: IEEE, 2013: 1-6.

[32] YOKOYA G, XIAO H, HATANAKA T. Multifactorial optimization using artificial bee colony and its application to car structure design optimization[C]//Proceedings of 2019 IEEE Congress on Evolutionary Computation (CEC). [S.l.]: IEEE, 2019: 3404-3409.

[33] LIU Y, MA L B, YANG G M. A survey on artificial bee colony algorithm[C]//Proceedings of the 7th Annual IEEE International Conference on Cyber Technology in Automation, Control and Intelligent Systems.

Hawaii, USA: IEEE, 2017: 1510-1515.

[34] KONG D P, CHANG T Q, DAI W J, et al. An improved artificial bee colony algorithm based on the ranking selection and the elite guidance[J]. Control and Decision, 2019, 34(4): 781-786.

[35] KARABOGA D, GORKEMLI B. A quick artificial bee colony (qABC) algorithm for optimization problems[C]//Proceedings of 2012 International Symposium on Innovations in Intelligent Systems & Applications. Trabzon, Turkey: IEEE, 2012: 1-5.

[36] WANG D, TAN D, LIU L. Particle swarm optimization algorithm: An overview[J]. Soft Computing, 2018, 22(2): 387-408.

[37] PONGCHAROEN P, HICKS C, BRAIDEN P M, et al. Determining optimum genetic algorithm parameters for scheduling the manufacturing and assembly of complex products[J]. International Journal of Production Economics, 2002, 78(3): 311-322.

Acknowledgements This work was supported by the Strategic Priority Research Program of the Chinese Academy of Sciences (No. XDA15040100), and the Youth Innovation Promotion Association of the Chinese Academy of Sciences (No.2021146). The authors would like to acknowledge the following people for their assistance: HU Tai, LIU Yurong, and GUO Guohang, all with the Laboratory of Satellite Operations Technology, Space Science Mission Operations Center, National Space Science Center.

Authors Ms. **YIN Xiaodan** received the B. E. degree in computer science and technology from Qufu Normal University, Shandong, China, in 2021. She is now a postgraduate at National Space Science Center and University of Chinese Academy of Sciences. Her research is focused on technology of satellite intelligent operation and control.

Prof. **BAI Meng** is a master supervisor in National Space Science Center, CAS. She obtained the M.S.E. degree from University of Chinese Academy of Sciences, Beijing, in 2012. She joined at National Space Science Center, CAS, in June 2012, where she is a senior engineer of Satellite Operations Technology Laboratory. Her research is focused on satellite intelligent operation and control technology, astronomy satellite intelligent scheduling and relevant fields.

Author contributions Ms **YIN Xiaodan** designed the study, complied the models, conducted the analysis, interpreted the results and wrote the manuscript. Prof. **BAI Meng** contributed to data and guided experimental work, checked experimental results. Mr. **LI Zhuoheng** contributed to orbital calculation. All authors commented on the manuscript draft and approved the submission.

Competing interests The authors declare no competing interests.

(Production Editor: ZHANG Bei)

超额订购下天文卫星短期任务的聚类规划方法

尹晓丹[1,2]，白　萌[1,2]，李卓恒[1]

(1.中国科学院国家空间科学中心卫星测运控技术实验室，北京 100190，中国；
2.中国科学院大学，北京 100049，中国)

摘要：当用户的观测需求超过卫星的观测能力时，天文卫星的任务规划就成为一个超额订购的问题。对于该问题，设计了一个包含聚类阶段和短期任务规划阶段的框架。首先建立了任务聚类模型，用于减少超额订购任务的规模。其次，使用聚类的结果作为输入，建立了短期任务规划的数学模型。最后，提出了一种自适应混合搜索策略的人工蜂群算法，在基本人工蜂群算法中引入了自适应精英全局-局部搜索策略和自适应变邻域最优搜索策略，以求解聚类和短期规划问题。所提出的算法在实验中表现出更好的寻优能力和更快的收敛速度。此外，它还有效地减少了聚类阶段的任务数量，提高了短期任务规划阶段的任务完成度。

关键词：天文卫星任务规划；超订购问题；任务聚类；短期任务规划；人工蜂群算法