Journal of Systems Engineering and Electronics
Vol. 34, No. 4, August 2023, pp.906 – 923

Mission scheduling of multi-sensor collaborative observation for space surveillance network

LONG Xi[1], CAI Weiwei[1,*], YANG Leping[1], and WANG Tianyu[2]

1. College of Aerospace Science and Engineering, National University of Defense Technology, Changsha 410073, China;
2. School of Traffic and Transportation Engineering, Central South University, Changsha 410073, China

Abstract: With increased dependence on space assets, scheduling and tasking of the space surveillance network (SSN) are vitally important. The multi-sensor collaborative observation scheduling (MCOS) problem is a multi-constraint and high-conflict complex combinatorial optimization problem that is non-deterministic polynomial (NP)-hard. This research establishes a sub-time window constraint satisfaction problem (STWCSP) model with the objective of maximizing observation profit. Considering the significant effect of genetic algorithms (GA) on solving the problem of resource allocation, an evolution heuristic (EH) algorithm containing three strategies that focus on the MCOS problem is proposed. For each case, a task scheduling sequence is first obtained via an improved GA with penalty (GAPE) algorithm, and then a mission planning algorithm (heuristic rule) is used to determine the specific observation time. Compared to the model without sub-time windows and some other algorithms, a series of experiments illustrate the STWCSP model has better performance in terms of total profit. Experiments about strategy and parameter sensitivity validate its excellent performance in terms of EH algorithms.

Keywords: multi-sensor observation, resource scheduling, sub-time window, evolution heuristic algorithm.

DOI: 10.23919/JSEE.2023.000104

1. Introduction

Space domain awareness (SDA) has been a topic of interest ever since the launch of Sputnik. With decreasing launch costs and the growth of the small satellite industry, the number of resident space objectives (RSOs) has grown significantly [1,2]. There are currently more than 4 000 active satellites in low-Earth orbit (LEO), and researchers estimate that by 2025 over 1 000 satellites can be launched each year [3]. The number of satellites may significantly exceed the current ability of SDA. Hence, in order to alleviate the burden from extreme increases of RSOs, we investigate the multi-sensor collaborative observation scheduling (MCOS) problem arising from the lack of sensor resources [4,5].

Generally, the space target has multiple time windows, and sensor resources need to observe the target within the time windows. Once an object is detected, a two-line element (TLE) set can be developed, and the RSOs can be tracked [6]. The tracking purpose involves determining and maintaining RSOs orbital parameters, according to which the current and future locations can be determined and predicted [7]. In this paper, we discuss the MCOS problem when the sensor resources observe targets contemplating which targets could be observed. Fortunately, although there is little research on the MCOS problem, tracking telemetry and command (TT&C) scheduling and other satellite resource scheduling can be useful references [8−11].

Fouad et al. developed a general framework for radar resource scheduling to allow scheduling flexibility and to handle multiple tasks using a single radar and then designed a taboo search heuristic algorithm [12]. Gao et al. proposed a multi-source heterogeneous sensor scheduling multiple objectives optimization model and used a multiple objective flexible fruit fly algorithm to solve it [13]. For the problem of scheduling time fragmentation, Gao et al. designed object time sensor coding and proposed individual feature cross and variation operations to prevent the results from being locally optimal [13]. Zhang et al. proposed an optimization model and a hybrid adaptive genetic algorithm (GA), then validated

Manuscript received December 22, 2021.
*Corresponding author.
This work was supported by the National Natural Science Foundation of China (11802333) and the Scientific Research Program of the National University of Defense Technology (ZK19-31).

their feasibility by using simulation experiments [14]. Kanit developed and compared novel scheduling models to a model reacting to the 2004 USSTRATCOM Strategic Directive 505-1 (SD 505-1) [15]. This scheduling model was developed to reduce time gaps between observations, prioritize high-value RSO, and retain maximum observation quality. Michael used parallel computation to optimize the geosynchronous (GEO) SDA [16]. He solved the problem of finding the most cost-effective combination for a high fidelity GEO SDA system and concluded that the optimal architecture had to change because of Earth-Sun angle variations. Yan et al. established a mathematical model with multiple constraints to solve the large-scale and complex scheduling problem of ground-based surveillance of space objects and constructed a hybrid algorithm with GA that used a simulated annealing algorithm for optimization [17]. They then discussed the rescheduling problem of ground-based space surveillance and proposed an ant colony algorithm to find the solution [18]. Along the same lines, Luo et al. considered multi-platform task co-allocation and established an oriented scheduling model of space objects using emergency observation among heterogeneous platforms, and this method was validated by simulation based on system tool kit (STK) and improved non-dominated sorting genetic algorithm (NSGA)-II algorithms [19]. For TT&C scheduling, there are some papers to be referenced [20–22]. Chen et al. established a mathematical model with the hybrid objective of maximizing profit as well as task completion rate [23]. They proposed a population perturbation and elimination strategy based on a GA and used simulation experiments to validate its effectiveness. Xue et al. presented the problem of multiple class TT&C resources using different networks to perform TT&C tasks jointly [24]. Then, they established the optimization model and proposed an improved GA. Meanwhile, deep learning has also been used to solve this problem [25].

However, most previous studies use an entire time window or arc segment as the scheduling basic unit, without considering the waste of resources when the real observation time is far less than this time window. In addition, RSOs tracking is accomplished quite frequently, and the sensor resources can observe many targets at the same time [15]. In order to solve the problem of MCOS, we propose a sub-time window constraint satisfaction problem (STWCSP) model and design an evolution heuristic (EH) algorithm. First, for comparative experiments, we design a general model without sub-time windows and solve it by the methods of first come first service (FCFS), improved FCFS (IFCFS), and a GA with penalty (GAPE). Second, we propose an STWCSP model and design an EH algorithm to solve it. Finally, we add simulation experiments to show the feasibility and efficiency of the proposed algorithm and the STWCSP model.

The rest of the paper is organized as follows. Section 2 introduces the process of scheduling, analyzes the existing problem, makes fundamental assumptions about the MCOS problem, and then establishes an STWCSP model. Section 3 describes the process of the EH algorithm, which includes a GAPE and heuristic rule. Section 4 adds the simulation environment and a series of experiments for testing the EH algorithm and the STWCSP model. Section 5 concludes the paper.

2. Preliminaries

2.1 Problem statement

The space surveillance network (SSN) currently tracks and maintains orbital information on objects in various orbits [16]. With limited resources and aging equipment, the current assets within the SSN’s inventory, comprising earth-based optical telescopes, space-based optical telescopes, and radar tracking sites, cannot keep up with an increasingly demanding mission.

This section mainly describes the MCOS problem and establishes the mathematical model. There are many targets and several ground sensors in the MCOS problem. Every target has a fixed state of motion, and each ground sensor has a fixed geographic location.

As shown in Fig. 1, the time window for each ground sensor to observe the target is limited, and the observation can only be completed within this time window. The flow of the time window calculation is shown in Fig. 2.

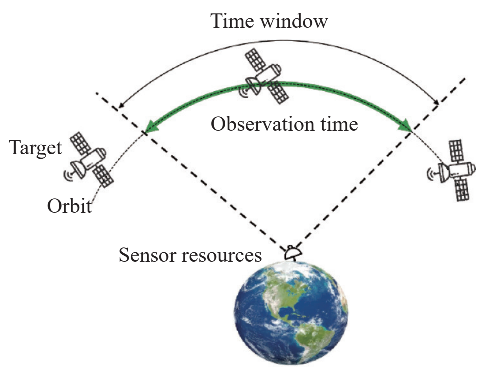


Fig. 1 Conventional observation scenario

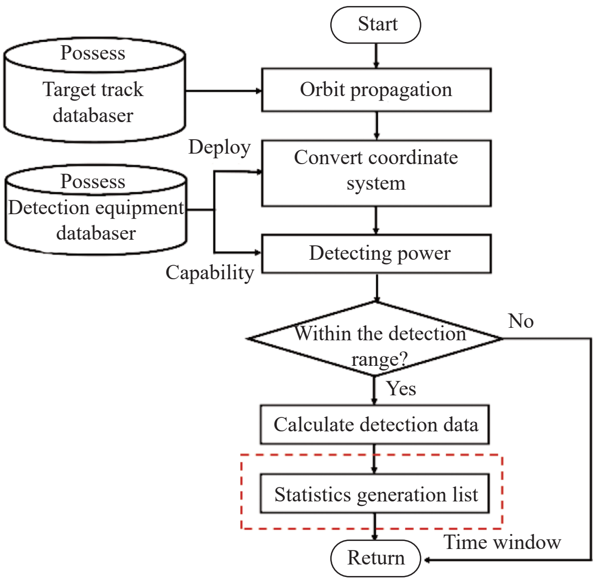


Fig. 2 Flow of time window calculation

Meanwhile, the antenna of the ground-sensor resources needs to point toward the space targets to complete the MCOS task. This means that after completing a current task, the same ground sensor needs to rotate the antenna to a specified angle before proceeding to the next task. Moreover, as shown in Fig. 3, because of the use of large power sensors, an antenna can observe multiple targets simultaneously.

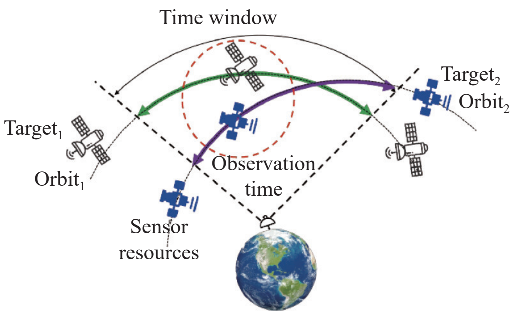


Fig. 3 Simultaneous observation of multi-targets

In addition, in our formulation of the MCOS problem, the real observation time is far less than the length of the time window. Taking previous studies [17–19] as an example, if only one target is observed in the whole time window, sensor resources will be wasted (as shown in Fig. 4). In this paper, a concept of sub-time windows is proposed, whose durations are the short actual observation times. In previous studies, the released time windows are abandoned immediately or not considered at all. In this paper, if the observation task is executed, the rest of the time in the time windows is released and it can be reused to generate new time windows. The start time of observation in this time window is obtained by an optimal algorithm. According to the above analysis, the outer result is which time window is used to observe the target, and the inner solution is the specific observation time, as shown in Fig. 4.

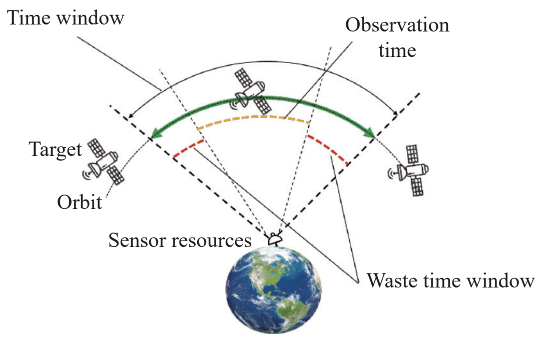


Fig. 4 Actual observation scenario

Overall, the MCOS problem's essence is to allocate sensor resources and execution time under various constraints such as visible time windows. This paper assumes that target priority, sensor resources, and mission requests are known before the scheduling algorithm is used.

We abstract and simplify the problem of MCOS, then make the following assumptions:

(i) The observation resource (sensor) is assumed to work generally throughout the scheduling cycle without damage.

(ii) The target does not maneuver during the scheduling period, and the time window does not change.

(iii) Once the observation starts execution, it cannot be allowed to be interrupted or preempted by other tasks.

2.2 STWCSP model

This section establishes the STWCSP model that contains the decision variables, constraints, and objective functions.

(i) Decision variables

To facilitate the concept of sub-time windows and reduce the complexity of traditional decision variables, this paper presents mixed decision variables ξ_{ijk} that combine Boolean and real number variables. ξ_{ijk} is an indicator variable that takes the value 1 when the target is observed in the kth rank of the ith target in its corresponding jth time window and takes the value 0 otherwise. Fig. 5 shows the schematic diagram of the decision variable.

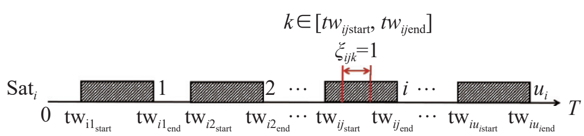


Fig. 5 Schematic diagram of decision variables

In Fig. 5, symbol tw is the time window and ξ_{ij} represents the ith target is observed in its jth time window, which is proposed by the conventional integer programming problem model [17]. Unlike existing studies, this paper introduces the real-number indexing k, taking values continuously between the relevant time window. Therefore, ξ_{ijk} is a mixed decision variable that contains both integer-only and real numbers, and it is also a vector defined in each time window as

$$\xi_{ijk} = [\xi_{ij1}, \xi_{ij2}, \cdots, \xi_{ijk}, \cdots, \xi_{ij\text{end\_time}}].$$

(ii) Constraints

The MCOS is a complex optimization problem involving various constraints. The constraints can be determined from targets and sensor resources.

For sensor resources, we can obtain the following constraints.

C1: The constraint of sensor observation distance. We assume the effective distance of the radar observation objective is limited, beyond which the target cannot be observed.

$$\forall i \in \mathbf{Res}, j \in \mathbf{Job}, R_{ij} \leqslant \mathbf{R}^i_{\max}$$

where **Res** represents the set of sensor resources, **Job** represents the set of targets, and $\mathbf{R}^i_{\max}$ represents the max observation distance of sensor i.

C2: The constraint of sensor minimum elevation angle. Only when the target is in the time window can the sensor observe it. Considering the ground is not horizontal, however, there may be high mountains and other obstacles that block the sensor limiting the effective angle,

$$\delta \geqslant \delta_{\min}$$

where δ is the elevation angle of sensor resources, and $\delta_{\min}$ is the minimum elevation angle.

C3: The constraint of sensor switching time. When the sensor conducts observation, the sensor needs a switching time during adjacent target observations. For the same sensor resource,

$$t^{\text{start}}_{q_{i+1}} - t^{\text{end}}_{q_i} \geqslant \gamma_q$$

where $t^{\text{start}}_{q_{i+1}}$ represents the $(i+1)$th start observation time for the qth sensor resources, $t^{\text{end}}_{q_i}$ is the ith end observation time for the qth sensor resources, and γ_q is the transfer time on the qth sensor resources.

C4: The constraint of sensor maximum observation capabilities. Only a certain number of targets can be observed simultaneously for one sensor resource. This number cannot exceed the upper limit of the sensor resources.

Assuming only m targets are observed at the same time, we have

$$\forall i \in \mathbf{Res}, j \in \mathbf{Job}, \sum_{\text{j}=1}^{m} t^i_{j\text{-goal}} \leqslant \text{Res}^i_{\text{track}}$$

where $t^i_{j\text{-goal}}$ represents the ith sensor resources accumulated time for the jth objectives, and $\text{Res}^i_{\text{track}}$ is the sensor i maximum observation capabilities.

For each objective, we obtain the following constraints.

C5: The constraint of the shortest observation time for each target. The shortest observation time influences the accuracy of orbital parameters and is related to the objectives' effective cross sections and radar resident times. If the radar parameter and target cross section are obtained for each target, the shortest observation time can be obtained as follows:

$$\text{tw}^{\text{end}}_{j\text{-true-time}} - \text{tw}^{\text{start}}_{j\text{-true-time}} = t^{\text{ob}}_j, \ \ \forall j \in \mathbf{Job}$$

where $\text{tw}^{\text{start}}_{j\text{-true-time}}$ represents the real start observation time for target j, $\text{tw}^{\text{end}}_{j\text{-true-time}}$ is the corresponding end time, and t^{ob}_j is the real observation time of objective j.

C6: The constraint of time windows. Only when the target is in the time window of the sensor can it be observed.

$$\begin{gathered}\forall \text{j} \in \mathbf{Job}, \\ [\text{tw}^{\text{start}}_{j\text{-true-time}}, \text{tw}^{\text{end}}_{j\text{-true-time}}] \subset [\text{tw}^{\text{start}}_{j\text{-time}}, \text{tw}^{\text{end}}_{j\text{-time}}]\end{gathered}$$

where $\text{tw}^{\text{start}}_{j\text{-time}}$ is the start time of the time window, and $\text{tw}^{\text{end}}_{j\text{-time}}$ is the corresponding end time.

C7: The constraints of observation frequency [26]. According to the characteristics of the target, the adjacent observation time of the same target cannot be too short. Observation frequency is varied for different targets over a scheduling period. When the number of observation for the same target is greater than 1, two adjacent observation times must be extended.

$$t_{\text{interval}} \geqslant \left[\frac{T}{N+1}, \frac{T}{N}\right]$$

where T represents one scheduling period, and N represents the number of times the target needs to be observed.

In the constraints described above, constraints C1 and C2 need to be considered when generating time window information, and constraints C3, C4, C5, C6, and C7 during scheduling.

(iii) Objective function

We take the total observation profit as the objective function defined as

$$\max f=\sum_{i=1}^{\text{Job}}\sum_{j=1}^{u_j}\sum_{k=\text{tw}_{i\text{-time}}^{\text{start}}}^{\text{tw}_{i\text{-time}}^{\text{end}}}\xi_{ijk}P_i$$

where P_i is the priority of target i. The larger P_i is, the more important target i is.

According to the above analysis, the scheduling model can be given as follows:

$$\text{find}\quad \xi_{ijk}$$

$$\max f=\sum_{i=1}^{\text{Job}}\sum_{j=1}^{u_j}\sum_{k=\text{tw}_{i\text{-time}}^{\text{start}}}^{\text{tw}_{i\text{-time}}^{\text{end}}}\xi_{ijk}P_i$$

s.t.

$$k\in[\text{tw}_{j\text{-time}}^{\text{start}},\text{tw}_{j\text{-time}}^{\text{end}}],$$

$$\text{tw}_{j\text{-true-time}}^{\text{end}}-\text{tw}_{j\text{-true-time}}^{\text{start}}=t_j^{\text{ob}},$$

$$[\text{tw}_{j\text{-true-time}}^{\text{start}},\text{tw}_{j\text{-true-time}}^{\text{end}}]\subset[\text{tw}_{j\text{-time}}^{\text{start}},\text{tw}_{j\text{-time}}^{\text{end}}],$$

$$\sum_{\text{j}=1}^{m}t_{j\text{-goal}}^{i}\leqslant \text{Res}_{\text{track}}^{i},$$

$$t_{q_{i+1}}^{\text{start}}-t_{q_i}^{\text{end}}\geqslant\gamma_q,$$

$$t_{\text{interval}}\geqslant\left[\frac{T}{N+1},\frac{T}{N}\right].$$

So far, the mathematical model of STWCSP about MCOS has been fully developed. However, because of the constraints mentioned above, and its higher-dimensional, NP-hard characteristics, it is almost impossible to find the optimal solution in practice. In many cases, it is also difficult to find a relatively good solution. Therefore, it is necessary and important to propose a specific algorithm for this problem.

3. Algorithm design

As an important part of bionic algorithms, GA has been widely used in various fields of combinatorial optimization [27–29]. However, GA cannot perform better in a mixed planning problem such as the one in this paper. To handle the MCOS problem better, we propose a method that consists of two parts. The first part (and the main framework) is GAPE and it is used in the outer optimization. By adaptively improving the flow of traditional GA genetic operators in the MCOS problem, the shortcomings of GA such as low efficiency and slow convergence can be overcome.

The second part is the mission planning algorithm (MPA), which is the inner optimal algorithm. If the outer task is executable, the start and the end time of it in the available time window can be determined by MPA. In the MPA, this paper takes the heuristic rules relating to preference, delay, and random strategies [30,31]. When the GAPE and MPA are terminated, the best solution in all the population is used as the final task sequence, and the final schedulable task sequence is obtained through an EH.

3.1 GAPE

Based on analysis and understanding of the MCOS problem, we optimize the population initialization, the coding method, selection operation, fitness function, crossover operation, mutation operation, and termination conditions. Standing on these improvements, we combine the heuristic rule with GAPE, and then propose an EH algorithm.

3.1.1 Population initialization

The process of population initialization aims to ensure the diversity and difference of individual. An absolutely random method of population initialization is utilized herein to ensure the significant differences among the gene fragments of each resulted individual. Therefore, the order of each schedule is random.

3.1.2 Coding method

In the coding method of GAPE, we use real number encoding, which shows the one-to-one correspondence between the task number and GAPE coding. As shown in Fig. 6, the coding method represents that the first task is executed in its second time window, the second task is completed in the corresponding jth time window,$\cdots$, the Nth task is performed in the last time window, etc. In the above encoding, the range of integer variables' va - lues depends on the number of corresponding time windows.

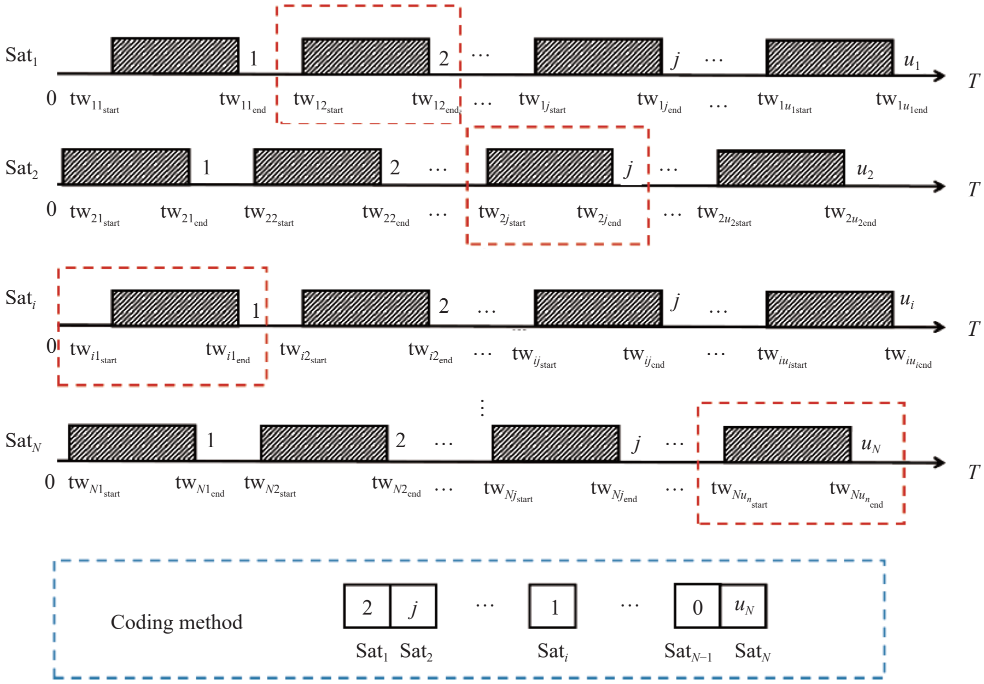


Fig. 6 Coding method

3.1.3 Fitness calculation

In the process of GAPE, fitness calculation plays an important role because it affects the probability of individuals' being selected to complete genetic operations. For closely combining the GAPE with the multi-sensor scheduling, this paper uses a previous model's section as the fitness function. The difficulty is that each individual task sequence should be input into the MPA (heuristic rule) to obtain an executable task scheme. The corresponding objective function value is calculated as the fitness of the current individual.

3.1.4 Selection

The selection should reflect the purpose. The individual that has a higher fitness can be selected with a high probability. In this paper, we take only the elite population to ensure the evolution process always increases in fitness.

3.1.5 Crossover

After selection, we compare a random number with crossover probability p_c and then use this to determine whether to perform the crossover operation on the current individual. In this paper, the coding method takes real numbers on upper optimization. Because of the real numbers, if two individuals are exchanged, then duplicate sequence fragments may occur for both individuals. To avoid this problem, this paper uses the crossover operation of two segments within an individual to maintain the feasibility of the solution.

As shown in Fig. 7, the whole length means that there are N tasks that need to be scheduled, and the p_c represents the crossover probability. When the crossover length is three, two break points containing the three alleles are randomly generated in two individuals. Then, the two segments are crossed, and two sequencesare obtained after the exchange. This is the completion of the crossover operation.

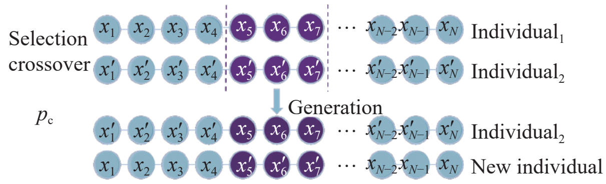


Fig. 7 Crossover operation

3.1.6 Mutation

After the above operations, we take the mutation operations to maintain the algorithm's feasibility. We design three mutation methods, which are shown in Fig. 8.

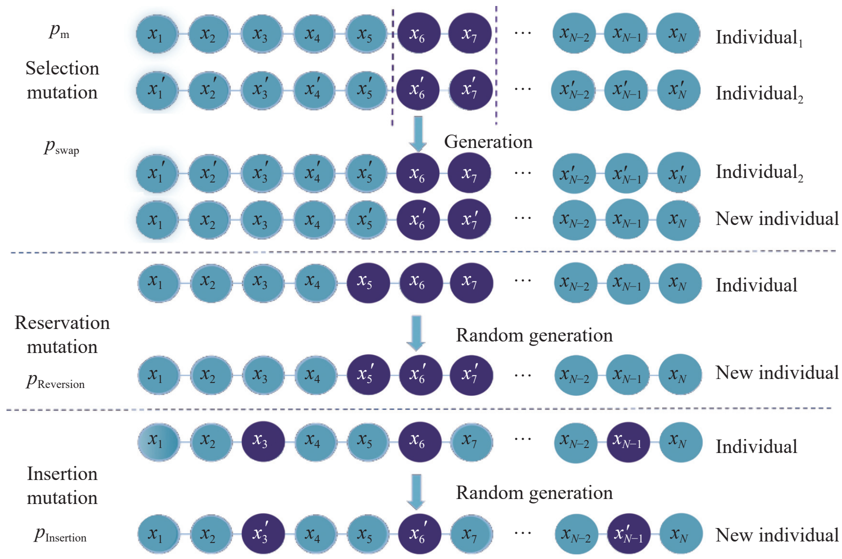


Fig. 8 Mutation operation

A random number between 0 and 1 is compared with the mutation probability p_m to determine whether to perform a mutation, but a mutation is less likely to occur than a crossover. Selection mutation is the same as selection crossover, and p_{Swap} parameter represents the selection mutation probability. In reversion mutation, continuous alleles on the chromosomes are randomly selected for mutation according to our constraints. The parameter $p_{\text{Reversion}}$ represents the probability of reservation mutation. In insertion mutation, discrete alleles on the chromosomes are randomly selected for mutation and the $p_{\text{Insertion}}$ represents the inversion probability. We must have: $p_{\text{Swap}} + p_{\text{Reversion}} + p_{\text{Insertion}} = 1$.

3.2 Heuristic rule

The heuristic rule is applied to arrange the specific execution time of the multi-sensor observation mission after the GAPE algorithm gets the new task scheduling sequence. In order to decrease calculation time, this paper uses the preference, delay, and random strategies [32,33]. The fitness value of the scheme is calculated based on the mutual selection of the task and the time window to determine whether the task is executed.

3.2.1 Preference strategy

Fig. 9 is the explanation of the preference strategy. This strategy means that if the time window is executed, the first target needs to be observed at the beginning of the window. The end time depends on the observation time, and the later missions are replenished in order. This method avoids the process of optimization and can significantly reduce the computational complexity.

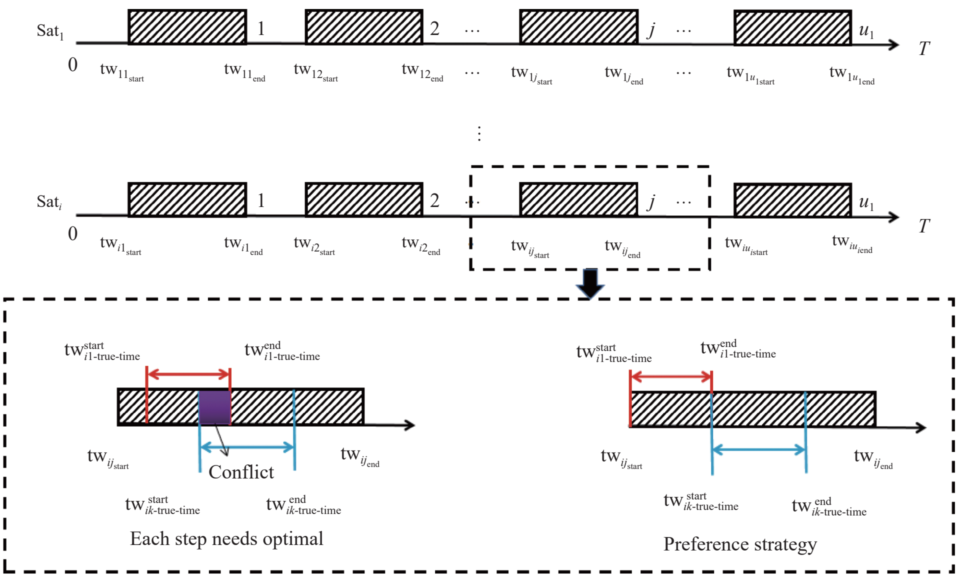


Fig. 9 Schematic diagram of preference strategy

The pseudo-code is labeled as Algorithm 1.

Algorithm 1 preference_strategy($\boldsymbol{S},\boldsymbol{W},\boldsymbol{C}$)

Input: $\boldsymbol{S},\boldsymbol{W},\boldsymbol{C}$ # $\boldsymbol{S}$ is the individual task, $\boldsymbol{W}$ is the set of available time windows, $\boldsymbol{C}$ is the set of constraints.

Output: Sol # Output solution

```
Sol = [] # is the set of executable scheduling solution
for i in S
  for j in W
    if S [i] meets the requirements of C
      Sol = S [i]
    End if
  End for
End for
sol = Sol[1]# select the first satisfies the constraints task
return sol
End procedure
```

3.2.2 Delay strategy

As shown in Fig. 10, the delay strategy is similar to the preference strategy, but it shows the opposite process. The first tasks start observation at the end of the time window.

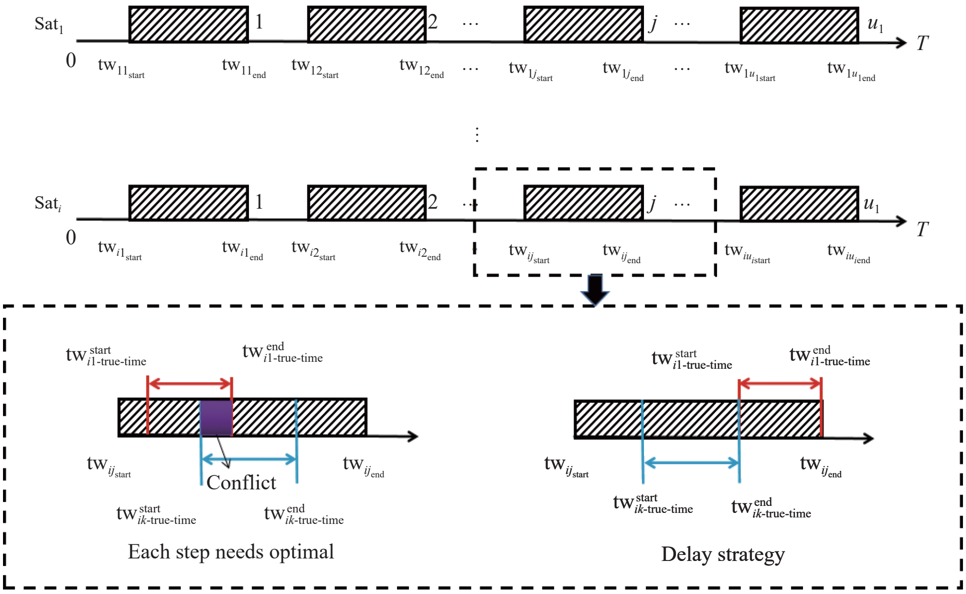


Fig. 10 Schematic diagram of delay strategy

The pseudo-code of the delay strategy is labeled as Algorithm 2.

Algorithm 2 delay_strategy($\boldsymbol{S},\boldsymbol{W},\boldsymbol{C}$)

Input: $\boldsymbol{S},\boldsymbol{W},\boldsymbol{C}$ # $\boldsymbol{S}$ is the individual task, $\boldsymbol{W}$ is the set of available time windows, $\boldsymbol{C}$ is the set of constraints.

Output: Sol # Output solution

```
Sol = [] # is the set of executable scheduling solution
for i in S
  for j in W
    if S [i] meets the require ments of C
      Sol = S [i]
    End if
  End for
End for
sol = Sol[len(Sol)]# select the last task
return sol
End procedure
```

3.2.3 Random strategy

Different from the preference and delay strategies, the random strategy emphasizes a random process as shown in Fig. 11. When the first target comes, it is randomly assigned for observation within the window. When the second target comes, it is also randomly assigned, but the two windows are allocated so that they do not overlap.

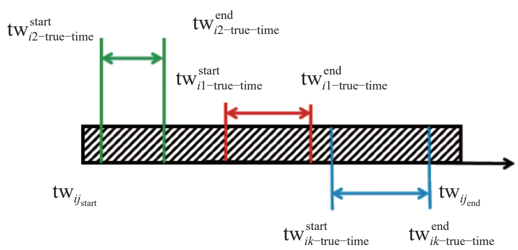


Fig. 11 Schematic diagram of random strategy

The pseudo-code of the random strategy is labeled as Algorithm 3.

Algorithm 3 random_strategy($\boldsymbol{S}, \boldsymbol{W}, \boldsymbol{C}$)

Input: $\boldsymbol{S}, \boldsymbol{W}, \boldsymbol{C}$ # $\boldsymbol{S}$ is the individual task, $\boldsymbol{W}$ is the set of available time windows, $\boldsymbol{C}$ is the set of constraints.
Output: Sol # Output solution

```
Sol = []# Sol is the set of executable scheduling solution
for i in S
   for j in W
      if S [i] meets the requirements of C
         Sol = S [i]
      End if
   End for
End for
sol = random(Sol)# select the random task
return sol
End procedure
```

3.3 EH algorithm

Subsections 3.1 and 3.2 introduce the GAPE and MPA algorithms, respectively. According to the problem analysis in Section 2, each target has many time windows in one scheduling period. We not only need to find the best time window for the sensors to observe the target, but also to determine the optimal observation moment within the corresponding time window. Taking N_R as the number of target, N_i as the number of time window of RSO$_i$, $\mathrm{tw}_{\mathrm{start}}^j$ and $\mathrm{tw}_{\mathrm{end}}^j$ as the start and end observation moment for time window j, the decision space could be $\prod_{i=1}^{N_R}\sum_{j=1}^{N_i}(\mathrm{tw}_{\mathrm{end}}^j-\mathrm{tw}_{\mathrm{start}}^j)$, which is extremely large for traditional algorithms. Moreover, the size of the decision space increases rapidly as the target scale increases. If the MPA is only used to solve this model, the computation time will decrease but could lead to a poor solution. If we only use the GAPE, this is a mixed-integer programming problem. There is no better solution algorithm, and almost all methods require a lot of calculation time.

Therefore, to improve the observation profit, we introduce the sub-time windows, and consider the calculation time and solution quality, to propose an EH algorithm. The EH algorithm is the combination of a GAPE and MPA (heuristic rule). In the outer optimization, this algorithm uses GAPE to get the time window in which a target is observed. Then, it uses MPA to get the specific task time. The rest of the time windows are released. After taking this operations, the outer decision space becomes $\prod_{i=1}^{N_R}N_i$, and the inner decision space becomes $\mathrm{tw}_{\mathrm{end}}^j-\mathrm{tw}_{\mathrm{start}}^j$. This splitting of the problem into upper and lower spaces decreases the size of total decision space. The pseudo-code is labeled as Algorithm 4.

Algorithm 4 evolution_heuristic($n, \boldsymbol{P}, t, n_{\mathrm{ep}}, w, p_m, p_{\mathrm{Swap}}, p_{\mathrm{Reservation}}, p_{\mathrm{Insertion}}, p_q, p_c$)

Input: $n, \boldsymbol{P}, t, n_{\mathrm{ep}}, w, \boldsymbol{S}$ # n: task number, $\boldsymbol{P}$: task priority, t: detection time, n_{ep}: number of equipment, w: time window
Output: t_{exe}, ep # Output the specific execution time of each task and the execution of each task by which equipment in the entire scheduling period.

```
W = random_gen_win(n) # generate windows for each tasks
S = find(W)# find time window for each target
target = cal_fitness(S, P)
foriinrange(Iterations) :
   select(P_q)
   crossover(P_q)
   mutation(p_m, p_Swap, p_Reservation, p_Insertion)
   update_population
   set[1] = preference_strategy(S, W, C)# Algorithm1
   set[2] = delay_strategy(S, W, C) # Algorithm2
set[3] = random_strategy(S, W, C) # Algorithm3
   target = cal_fitness(S, P)
   if population individual better than the global
      globalbest = popindividual
   End if
   Record(globalbest) # generate the best generation
End for
Cal_total_benefits()
gen_sche_list()
End procedure
```

Fig. 12 shows the process of the EH. The basic parameters of this algorithm are input at the execution of the program, including the task number, task priority, observation time, equipment number, time window, crossover probability, and variation probability. Then, it generates the initial population and defines the execution time window for each target. Later, based on the above solution and calculating the fitness by Algorithm 1, Algorithm 2, and Algorithm 3, and the best individual for each iteration can be defined. If the above procedure is within the algorithm iteration, the optimal individual and the elite population are retained in each generation. The remaining individuals perform crossover and mutation operations, and the specific process of cross variation is detailed in Subsection 3.1. After finishing the algorithm, the best solution can be obtained.

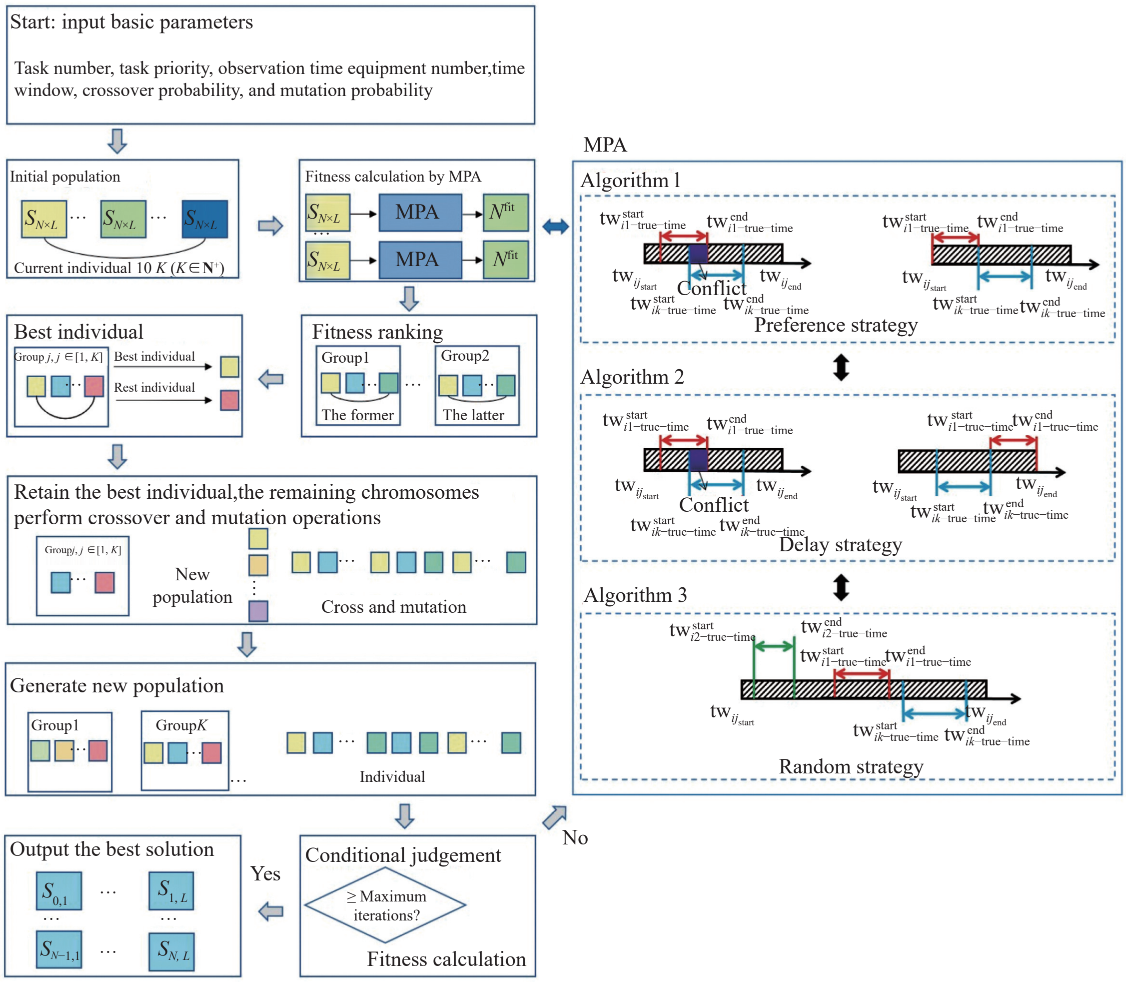


Fig. 12 Process of the EH algorithm

4. Numerical simulation

In this section, we design a series of comparative experiments to investigate the performance of models in various aspects thoroughly.

4.1 Experimental design

(i) Experimental instances

This paper takes three radars as the sensor resources. The position of sensors are (40.038 6°N,75.596 6°W), (30.038 6°S,15.596 6°E), and (10.038 6°S,105.597°E), respectively. The sensors' transfer time is 5 s, 100 s, and 80 s. The task size of test instances ranges from 500 to 1 300, and the relevant orbital database can be obtained from the well-known CelesTrak website. Each target priority ranges from 1 to 10, and the observation time ranges from 200 s to 500 s.

(ii) Comparison algorithms

This paper selects three algorithms for comparison experiments, and these algorithms all optimize the model without sub-time windows. The first algorithm is FCFS, and the second is IFCFS. Compared to FCFS, IFCFS adds a cycle policy, and generally achieves a better solution with a higher total priority, although it is a heuristic algorithm. The third is GAPE, which optimizes integer time window planning. The EH algorithm contains GAPE preference strategy (GAPE-HP), GAPE delay strategy (GAPE-HD), and GAPE random strategy (GAPE-HR).

(ii) Parameter settings

Default parameters of GAPE are shown in Table 1, and the parameters of GAPE-HP, GAPE-HD, and GAPE-HR are the same as those involved in GAPE.

Table 1 GAPE parameters

Parameter	Value
Gen	500
NP	200
p_c	0.8
p_m	0.1
p_{Swap}	0.2
$p_{\mathrm{Reversion}}$	0.5
$p_{\mathrm{Insertion}}$	0.3

4.2 Experiments on different target scales with different algorithms

In order to validate the performance of the proposed algorithms, we first conduct some experiments that include nine scenarios of different target scales ranging from 500 to 1300. The comparison experiments are carried out by using the three comparison algorithms mentioned in Subsection 4.1, and the final results are shown in Table 2.

Table 2 Experimental results of different algorithms at different target scales

Target scale	Model without sub-time windows			STWCSP model			EH average result
	FCFS	IFCFS	GAPE	GAPE-HP	GAPE-HD	GAPE-HR	
500	518	2257	2449	2515	2496	2554	2521.67
600	554	2307	2760	2882	2974	3024	2960
700	595	2348	3011	3270	3314	3317	3300.33
800	595	2369	3199	3674	3498	3512	3561.33
900	612	2389	3479	3927	4038	4149	4038
1000	640	2399	3726	4142	4135	4233	4170
1100	642	2402	3885	4409	4583	4373	4455
1200	656	2425	4073	4631	4617	4635	4627.67
1300	665	2449	4231	4858	4782	4801	4813.67

As can be seen from Table 2, when the scale of the target increases, the total task priority also increases. Furthermore, the EH algorithm proposed herein can solve the STWCSP model and obtain a better solution compared with the conventional methods, regardless of the number of targets. Fig. 13 represents the observation profit of different algorithms at different target scales, and it shows that the STWCSP model and the EH algorithm are better than conventional methods.

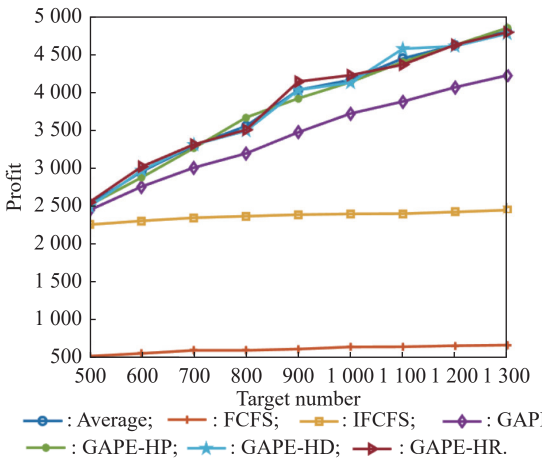


Fig. 13 Comparison of profit between EH and conventional algorithms at different target scales

In addition, as shown in Fig. 14, when the target scale increases, the performance of the EH algorithm becomes better compared to the conventional algorithm. Although profit difference decreases when the target scale is 1000 and 1200, this is because the EH algorithm is a random search algorithm, and is not guaranteed always to find the global optimum solution. Overall, however, the nine curves increase on average over the tested domain of the target scale.

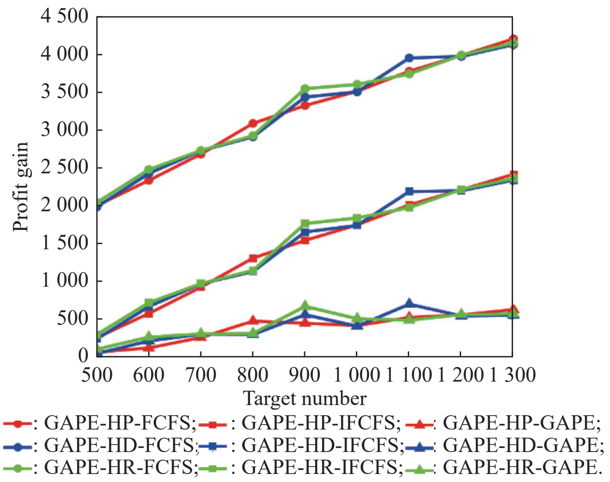


Fig. 14 Profit gain of EH algorithm with increasing target scale

Fig. 15 shows the evolution process of the GAPE and EH algorithms. From the curve comparison of EH and GAPE in this figure, EH has evidently outperformed all tests, but the three strategies of EH does not have significant characteristic. However, in this paper’s test, when the target scale is small, the GAPE-HR might perform better than GAPE-HP and GAPE-HD.

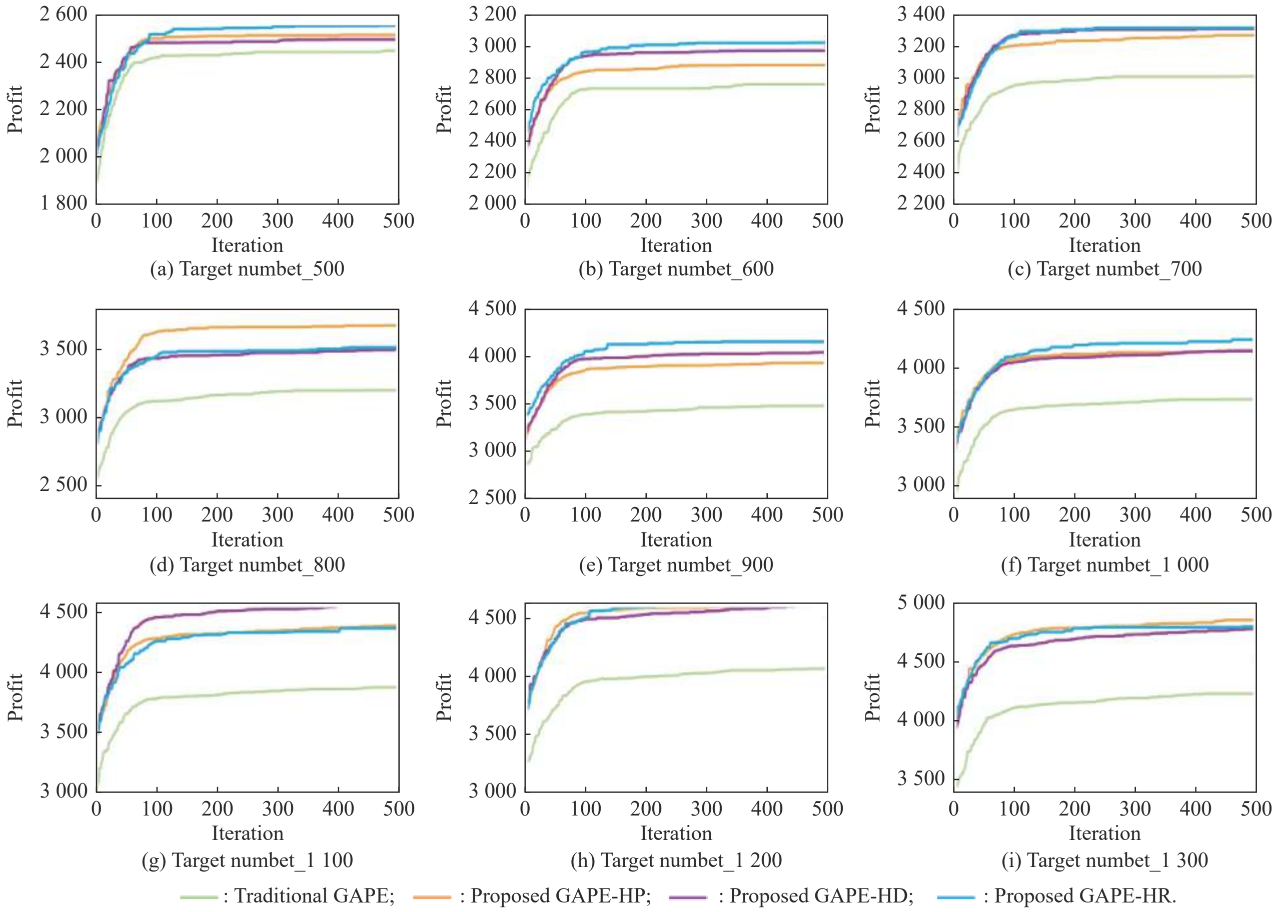


Fig. 15 Historical best profit of different algorithms at different target scales versus iterations

When the target scale is large, the three strategies result in similar profit values.

4.3 Experimental results on different sensor observation capabilities

To validate the effect of different sensor observation capabilities on the EH algorithm and STWCSP model, we conduct experiments on instances that include nine scenarios. In this part, we assume that each sensor has a different observation capabilities. Combining this with the simulation data presented in this paper, we assume that the number of targets observed by different sensors at the same time are (1 1 1), (1 1 5), (1 5 1), (1 5 5), (2 2 2), (3 3 3), (4 4 4), (5 1 1), and (5 5 5). The other parameters are the same as in Subsection 4.1. Table 3 presents the final results under different sensor observation capabilities.

Table 3 Experimental results with different sensor observation capabilities

Capability	GAPE	GAPE-HP	GAPE-HD	GAPE-HR
(1 1 1)	2128	2676	2660	2516
(1 1 5)	3317	3699	3709	3742
(1 5 1)	3797	4283	4380	4268
(1 5 5)	4939	4874	4645	5015
(2 2 2)	4231	4808	4898	4878
(3 3 3)	4838	5226	5251	5257
(4 4 4)	5108	5339	5358	5389
(5 1 1)	3726	4142	4135	4233
(5 5 5)	5350	5406	5414	5420

As shown in Table 3, if the sensor has bigger observation capabilities, the number of the observed targets is larger. The regulation can be obtained in Fig. 16.

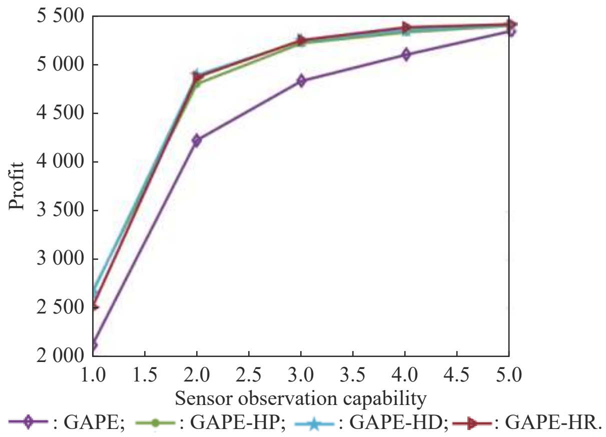


Fig. 16 Influence of sensor observation capabilities on profit

In addition, we also find that when the total number of sensor observation capabilities is the same but each sensor has a different number of observations, the solutions obtained by all types of GAPE are relatively similar, but those solved by EH are better than by all types of GAPE.

Fig. 17 is the historical best profit of each different algorithm for each iteration with different sensor observation abilities, from the curve comparison of EH and GAPE in this figure, EH evidently outperforms all tests, but the fourth sub-figure shows different characteristics. In this result, the GAPE-HP and GAPE-HD algorithm both are lower than GAPE. In addition, when all sensors have an observation capability of 5, the evolutionary curve quickly converges to the optimal solution, and all tasks are completed. This is because all targets can be observed when the sensor has a higher observed ability.

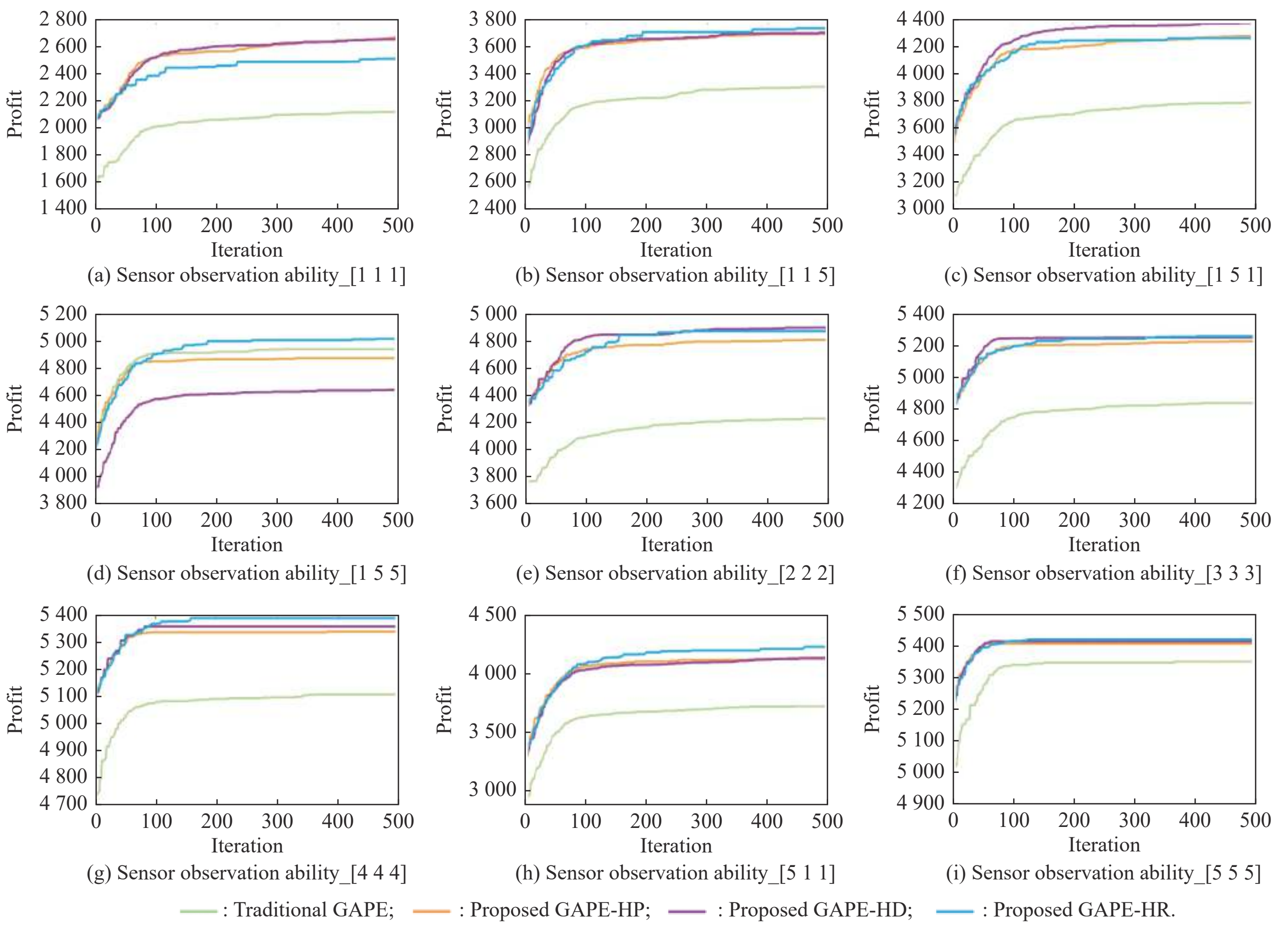


Fig. 17 Historical best profit of different algorithms with different sensor observation abilities versus iterations

From the above analysis, we can see that different sensor capabilities can cause different solutions. In future work, we aim not only to develop the scheduling algorithm of the multi-sensor problem but also to study sensor performance as well.

4.4 Experiments on target observation time

This section discusses the different target observation time's influence on the EH algorithm and STWCSP model. We conduct a nine-scenarios comparison that includes 0 to 900 s. The ranges of observation time are [0 100], [100 200], [200 300], [300 400], [400 500], [500 600], [600 700], [700 800], and [800 900]. Table 4 shows the results with different target observation time's influence on the solution.

Table 4 Experimental results with different target observation time

Observation	GAPE	GAPE-HP	GAPE-HD	GAPE-HR
[0 100]	3762	4998	4963	4993
[100 200]	3765	4756	4617	4586
[200 300]	3813	4430	4394	4448
[300 400]	3740	4193	4237	4184
[400 500]	3705	3828	3994	3797
[500 600]	3826	3709	3715	3558
[600 700]	3708	3468	3419	3343
[700 800]	3784	3186	3182	3705
[800 900]	3783	2982	2898	2951

As shown in Table 4, we know that the target observation time has little influence on the GAPE algorithm and traditional model, but it has a significant influence on the EH algorithm and STWCSP model. The specific relationship between the STWCSP model and EH algorithm's solution is shown in Fig. 18.

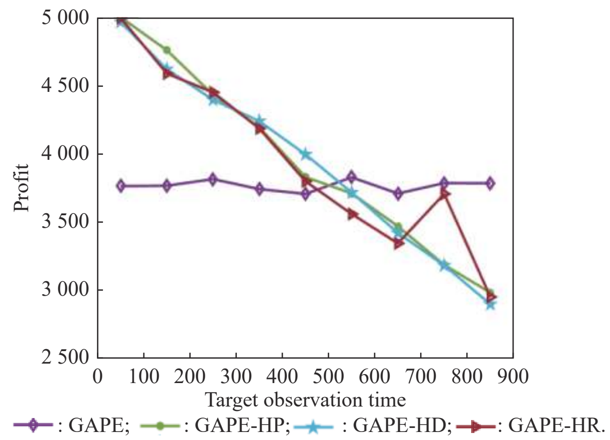
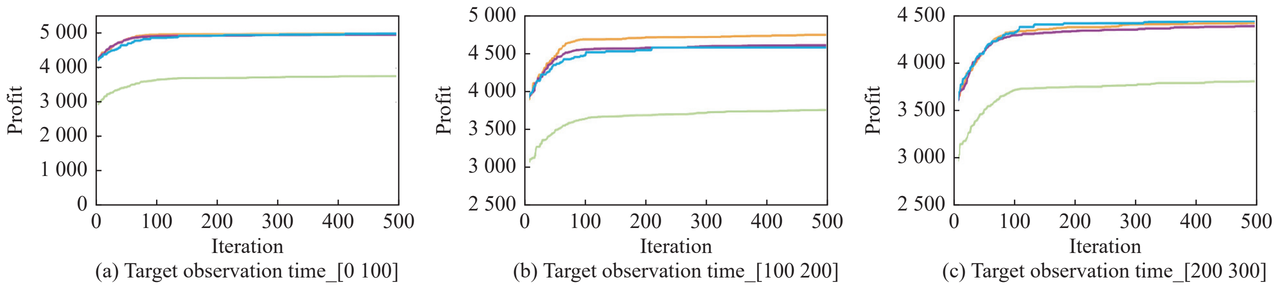


Fig. 18 Influence of observation time on profit

Fig. 18 shows the total observation profit change curves of different observation time for the different algorithms. Here we can see the different target observation times have little influence on the GAPE. However, the same cannot be said of the EH algorithm. When the target observation time becomes larger, the profit of task completion decreases. From the above analysis, we conclude that in the STWCSP model and the EH algorithm proposed, these methods do not apply if the target observation time is very long. In practice, however, some targets need longer observation time, while some need shorter observation time. Fortunately, the method presented in this case still works (Subsection 4.2) universally.

Fig. 19 shows the historical best profit of different algorithms for each iteration at different target observation time. From this figure, we can see that the models and algorithms designed are more efficient when the target observation time is short. When the target observation time is long enough, however, the traditional models and algorithms are more effective.



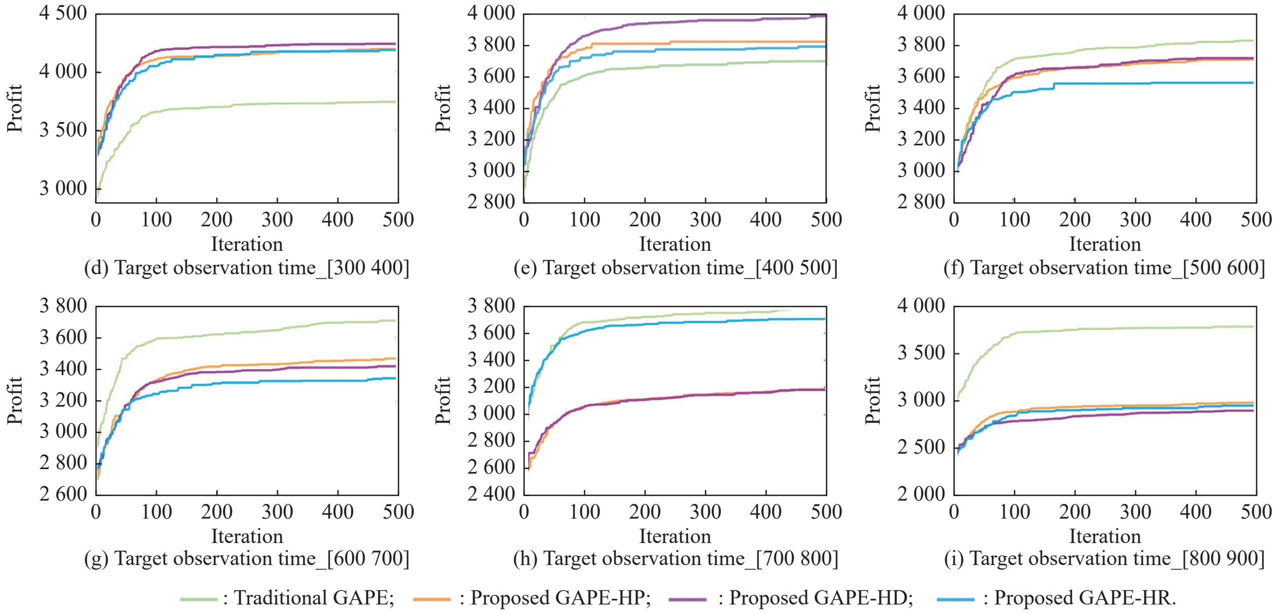


Fig. 19 Historical best profit of different algorithms with different target observation time versus iterations

Specifically, the traditional model performs better when the target observation time exceeds 500 s. In practice, the target is randomly distributed, and each target observation time is determined by its cross-section, which is generally more random, and hence our proposed model and methods will be more effective compared to existing models than in our experiments (Subsection 4.2).

As mentioned above, the models and algorithms proposed are not applicable when the target observation time is long, but in practice, the cross-section of each target varies greatly and it results in that the observation time for each target varies much.

4.5 Experimental results on different sensor transfer time

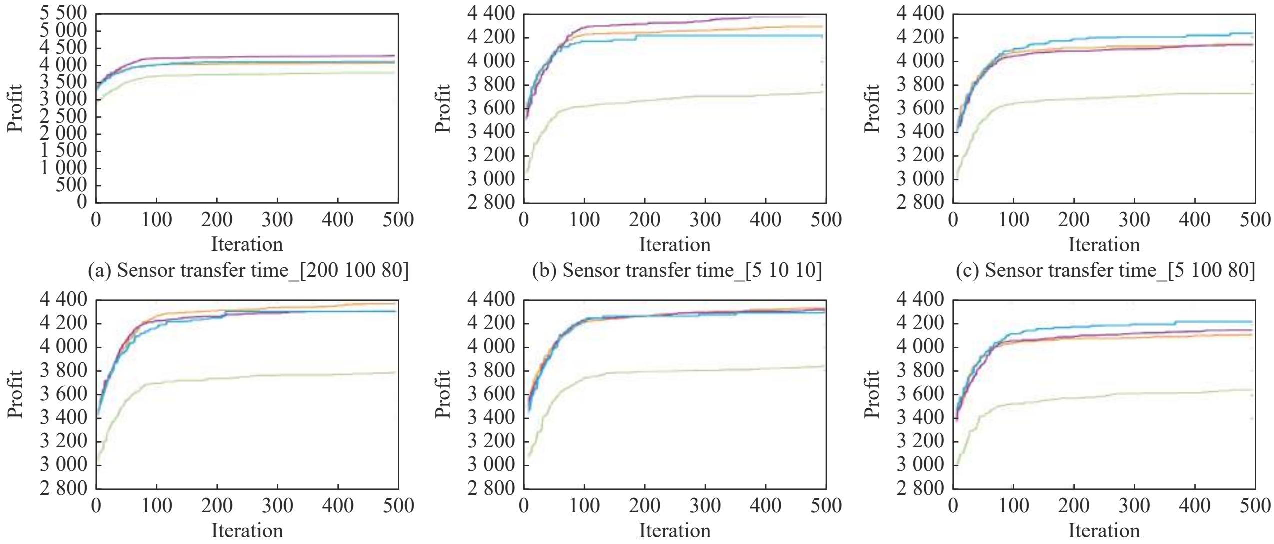
To measure the influence that different sensor transfer time have on the STWCSP model and EH algorithm, this section conducts six experiments where the sensor transfer time are [20 100 80], [5 10 10], [5 100 80], [5 25 20], [5 50 40], and [50 100 80]. The other parameters are the same as in Subsection 4.1. We use GAPE, GAPE-HP, GAPE-HD, and GAPE-HR to solve these six problems, and the results are shown in Table 5.

Table 5 Experimental results with different sensor transfer time

Transfer time	GAPE	GAPE-HP	GAPE-HD	GAPE-HR
[20 100 80]	3802	4080	4293	4119
[5 10 10]	3740	4293	4380	4218
[5 100 80]	3726	4142	3650	4233
[5 25 20]	3788	4368	4304	4302
[5 50 40]	3842	4329	4316	4294
[50 100 80]	3645	4110	4149	4219

As shown in Table 5, we see that the EH algorithm and STWCSP model proposed here are better than the traditional methods regardless of sensor transfer time. Furthermore, the transfer time of different sensors has some effects on the final profit. In the experiments presented here, we see that total profit increases if sensor transfer time decreases slightly.

Fig. 20 shows the historical best profit of different algorithms for each iteration at different sensor transfer time. We can see that the proposed model and algorithms all perform better than the model without sub-time windows. Additionally, in this experiment, we find that the total magnitude of sensor transfer time is little related to the total profit.



(a) Sensor transfer time_[200 100 80]
(b) Sensor transfer time_[5 10 10]
(c) Sensor transfer time_[5 100 80]
(d) Sensor transfer time_[5 25 20]
(e) Sensor transfer time_[5 50 40]
(f) Sensor transfer time_[50 100 80]

: Traditional GAPE; : Proposed GAPE-HP; : Proposed GAPE-HD; : Proposed GAPE-HR.

Fig. 20 Historical best profit of different algorithms with different sensor transfer time versus iterations

4.6 Experimental results on EH sensitivity validation

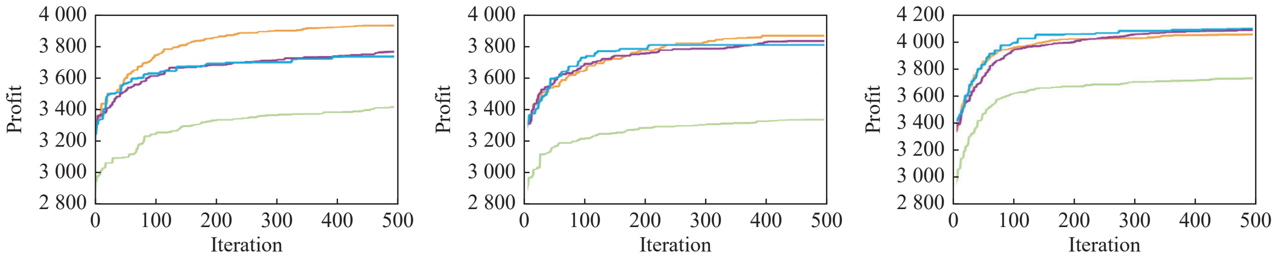
To explore the effect of experimental parameters on the results thoroughly, we carry out extensive experiments of the parameter sensitivity validation of EH. In this part, we set up six experiments over the parameters $p_c, p_m, p_{\rm Swap}, p_{\rm Reversion}$, i.e., [0.5 0.4 0.3 0.3], [0.5 0.5 0.4 0.4], [0.7 0.2 0.3 0.4], [0.8 0.1 0.2 0.5], [0.8 0.1 0.3 0.4], and [0.9 0.1 0.4 0.3]. Here, the sensor transfer time is [50 80 100], the target observation time is randomly distributed from 200 s to 500 s, and the sensor observation ability is [5 1 1]. The model is solved by different algorithms, and the final result are shown in Table 6.

Table 6 Experimental results with different algorithm parameters

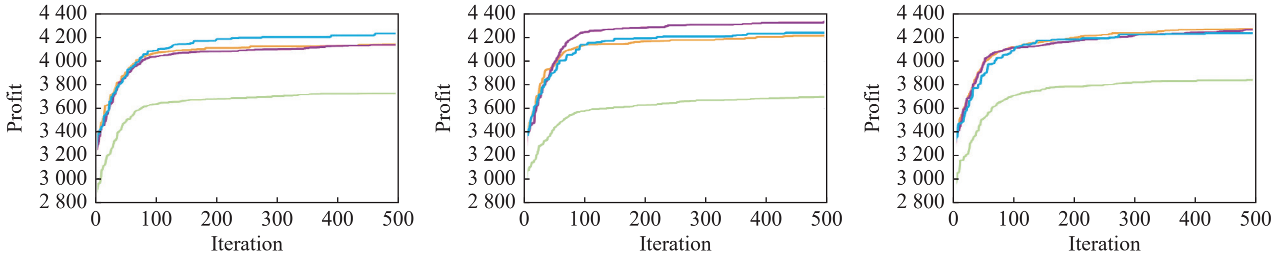
Probability	GAPE	GAPE-HP	GAPE-HD	GAPE-HR
[0.5 0.4 0.3 0.3]	3413	3929	3765	3732
[0.5 0.5 0.4 0.4]	3341	3872	3838	3813
[0.7 0.2 0.3 0.4]	3735	4059	4093	4105
[0.8 0.1 0.2 0.5]	3726	4142	4135	4233
[0.8 0.1 0.3 0.4]	3700	4214	4334	3722
[0.9 0.1 0.4 0.3]	3846	4277	4271	4242

As shown in Table 6, we can see that changing the parameters of the EH algorithm hardly influences the total task profit. The performances of our proposed STWCSP model and EH algorithm are all better than that of the model without sub-time windows.

Fig. 21 shows the historical best profit of different algorithms for each iteration at different algorithm parameters. When the crossover probability is high, a better result can be obtained, but this influence is small. Additionally, if the crossover probability is in a certain interval, the other parameters have little effect on the results.



(a) p_c=0.5, p_m=0.4, $p_{\rm Swap}$=0.3, $p_{\rm Reversion}$=0.3
(b) p_c=0.5, p_m=0.5, $p_{\rm Swap}$=0.4, $p_{\rm Reversion}$=0.4
(c) p_c=0.7, p_m=0.2, $p_{\rm Swap}$=0.3, $p_{\rm Reversion}$=0.4



(d) p_c=0.8, p_m=0.1, p_{Swap}=0.2, $p_{\text{Reversion}}$=0.5 (e) p_c=0.8, p_m=0.1, p_{Swap}=0.3, $p_{\text{Reversion}}$=0.4 (f) p_c=0.9, p_m=0.1, p_{Swap}=0.4, $p_{\text{Reversion}}$=0.3

: Traditional GAPE; : Proposed GAPE-HP; : Proposed GAPE-HD; : Proposed GAPE-HR.

Fig. 21 Historical best profit of different algorithms with different algorithm parameters versus iterations

5. Conclusions

As the number of RSOs increases, enhancing SDA capabilities for detecting, cataloging, and tracking RSO becomes more critical for sustaining long-term space operations. In this paper, taking three ground sensors for example, we study the problem of their optimal daily scheduling. The main contributions are as follows:

(i) The concept of sub-time window is proposed, and an MCOS STWCSP model which avoids the waste of sensor resources is established.

(ii) The algorithms are tested in a series of scenarios. According to the results of the simulation, our proposed algorithm has a better performance in total profit compared with GAPE, FCFS, and IFCFS.

(iii) The influence of target observation time, sensor transfer time, sensor observation capabilities, and algorithm parameters on our proposed STWCSP model and EH algorithm is analyzed.

However, there are still some issues for further research, such as the scheduling method in emergency situations and the resource constraint reasoning methods of MCOS, which will be addressed in the future.

References

[1] GUIDO P, DAVID V C, MARIA G D V, et al. SPOOK: a tool for space objects catalogue creation and maintenance supporting space safety and sustainability. Acta Astronautica, 2021, 188: 89–98.

[2] HU Y P, LI K B, LIANG Y G, et al. Review on strategies of space-based optical space situational awareness. Journal of Systems Engineering and Electronics, 2021, 32(5): 1152–1166.

[3] THOMAS G R, PENG M S, DANIEL J. A deep reinforcement learning application to space-based sensor tasking for space situational awareness. Proc. of the Advanced Maui Optical and Space Surveillance Technologies Conference, 2021: 1−13.

[4] BRYAN D L, CAROLIN E F. Space situational awareness sensor tasking: comparison of machinelearning with classical optimization methods. Journal of Guidance, Control, and Dynamics, 2020, 43(2): 262–273.

[5] NICHOLAS R C, BRANDON A J. Risk-aware sensor scheduling and tracking of large constellations. Advance in Space Research, 2021, 68(6): 2530–2550.

[6] TAO X F, LI Z, XU C, et al. Track-to-object association algorithm based on TLE filtering. Advance in Space Research, 2021, 67(8): 2304–2318.

[7] SAN M M, VERGARA E P, PEREA I, et al. Hybrid SGP4 orbit propagator. Acta Astronautica, 2017, 137: 254–260.

[8] WANG Y X, ZHOU D, SONG N B, et al. Concurrent reconfiguration of resource-oriented emergency TT&C mission planning for space information networks. Journal of Communications and Information Networks, 2021, 6(2): 142–152.

[9] LI C D, XU W, XU L, et al. An approach to multi-satellite TT&C resource scheduling based on multi-agent technology and comprehensive weighted priority determination method. Journal of Physics: Conference Series, 2021, 1812: 012001.

[10] WU J, CHEN Y N, HE Y M, et al. Survey on autonomous task scheduling technology for Earth observation satellites. Journal of Systems Engineering and Electronics, 2022, 33(6): 1176–1189.

[11] CHANG Z X, ZHOU Z B, YAO F, et al. Observation scheduling problem for AEOS with a comprehensive task clustering. Journal of Systems Engineering and Electronics, 2021, 32(2): 347–364.

[12] FOUAD B A, HASAN M. An optimization model and tabu search heuristic for scheduling of task on a radar sensor. IEEE Sensors Journal, 2016, 16(17): 6695–6702.

[13] GAO J L, XING Q H, LIANG Z B. Multiple sensor resource scheduling model and algorithm for high speed target tracking in aerospace. Systems Engineering and Electronics, 2019, 41(10): 2244–2250. (in Chinese)

[14] ZHANG H W, XIE J W, GE J A, et al. A hybrid adaptively genetic algorithm for task scheduling problem in the phased array radar. European Journal of Operational Research, 2019, 272(3): 868–878.

[15] KANIT D. Comparison of novel heuristic and integer programming schedulers for the USAF space surveillance network. Ohio, America: Air Force Institute of Technology, 2019.

[16] MICHAEL S F. Optimization of geosynchronous space situational awareness architectures using parallel computation. Ohio, America: Air Force Institute of Technology, 2018.

[17] YAN Q Q, SHEN H R, SHAO Q L. Space object ground-based surveillance scheduling based on genetic-simulated annealing algorithm. Systems Engineering and Electronics, 2015, 37(12): 2764–2771. (in Chinese)

[18] YAN Q Q, SHEN H R, SHAO Q L. Ground-based space surveillance rescheduling based on ant colony optimization. Ordnance Industry Automation, 2016, 35(3): 1–5.

[19] LUO J, YU X H, WANG J J. Cooperative scheduling prob-

lem space target observation of multi-platform. Fire Control & Command Control, 2021, 46(12): 81–87.
[20] GU X S, BAI J, ZHANG C A, et al. Study on TT&C resources scheduling technique based on inter-satellite link. Acta Astronautica, 2014, 104(1): 26–32.
[21] ZHANG Z J, HU F N, ZHANG N. Ant colony algorithm for satellite control resource scheduling problem. Applied Intelligence, 2018, 48(10): 3295–3305.
[22] XU Y, JIAO Y Y, PAN X G, et al. An efficient scheduling method for satellite TT&C resources. Proc. of the 7th International Conference on Big Data and Information Analytics, 2021: 340−349.
[23] CHEN M, WEN J, SONG Y J, et al. A population perturbation and elimination strategy based genetic algorithm for multi-satellite TT&C scheduling problem. Swarm and Evolutionary Computation, 2021, 65: 100912.
[24] XUE N Y, DING D, WANG H M, et al. Multi-type TT&C resource scheduling method based on improved genetic algorithm. Systems Engineering and Electronics, 2021, 43(9): 2535–2543. (in Chinese)
[25] JIANG F B, WANG K Z, DONG L, et al. Deep-learning-based joint resource scheduling algorithms for hybrid MEC Networks. IEEE Internet of Things Journal, 2020, 7(7): 6252–6265.
[26] ANDREW J S, ALAN L T. Optimal linear orbit determination. Journal of Guidance, Control, and Dynamics, 2020, 43(3): 628–632.
[27] LI Y, WU H L, SUN Y H. Improved adaptive genetic algorithm based RFID positioning. Journal of Systems Engineering and Electronics, 2022, 33(2): 305–311.
[28] MATTEW S, TAO X H, SOMAN E, et al. A novel genetic algorithm based system for the scheduling of medical treatments. Expert Systems with Applications, 2022, 195: 116464.
[29] ZHANG J W, LIU W J, LIU W L. An efficient genetic algorithm for decentralized multi-project scheduling with resource transfers. Journal of Industrial and Management Optimization, 2022, 18(1): 1–24.
[30] MEHRDAD A, YAHIA Z M, ALI M. A rule-based heuristic algorithm for on-line order batching and scheduling in an order picking warehouse with multiple pickers. RAIRO-Operation Research, 2020, 54(1): 101–107.
[31] SRINATH N, YILMAZLAR I O, KURA M E, et al. Introducing preferences in scheduling applications. Computer & Industrial Engineering, 2022, 163: 10783 1.
[32] OSMAN H T, DAVID A, EKREM M. Comparison of heuristic priority rules in the solution of the resource-constrained project scheduling problem. Sustainability, 2021, 13(17): 9956.
[33] LIANG J, ZHU Y H, LUO Y Z, et al. A precedence-rule-based heuristic for satellite onboard activity planning. Acta Astronautica, 2021, 178: 757–772.

Biographies

LONG Xi was born in 1999. He received his B.S. degree from Shenyang Aerospace University (SAU), Shenyang, China, in 2020. He is a Ph.D. student with the College of Aerospace Science and Engineering, National University of Defense Technology (NUDT). His research interest is space domain awareness technique.
E-mail: longxi_1999@163.com

CAI Weiwei was born in 1987. He received his B.S. and Ph.D. degrees from National University of Defense Technology (NUDT), Changsha, China, in 2009 and 2015, respectively. He is a lecturer with the College of Aerospace Science and Engineering, NUDT. His research interests include space mission planning, dynamics and control of on-orbit operation.
E-mail: caiweiwei@nudt.edu.cn

YANG Leping was born in 1964. He received his B.S. and M.S degrees from National University of Defense Technology (NUDT), Changsha, China, in 1984 and 1987, respectively. He is a professor with the College of Aerospace Science and Engineering, NUDT. His research interests include space mission planning, dynamics and control of on-orbit operation.
E-mail: ylp_1964@163.com

WANG Tianyu was born in 1998. He received his B.S. degree from Taiyuan University of Technology (TYUT), Taiyuan, China, in 2020. He is an M.S. student with the School of Traffic and Transportation Engineering, Central South University (CSU). His research interests include theory and application of intelligent planning.
E-mail: wangtianyu189@163.com