

Article

A Resource Scheduling Algorithm for Optical Ground Stations Supporting Satellite Observation Tasks

Shijie Zhai, Wenhua Cheng  and Tinghua Zhang *

Department of Electronic and Optical Engineering, Space Engineering University, Beijing 101416, China; 15810887677@163.com (S.Z.); cwhcn@foxmail.com (W.C.)

* Correspondence: zth-gd@163.com

Abstract

To address the resource scheduling problem of ground-based optical stations for key space target observation, this study proposes a station-structured genetic algorithm driven by dual-level tabu memory. The proposed method integrates station-grouped encoding, heuristic initialization, station-structured crossover, neighborhood-enhanced mutation, and local and global tabu-memory mechanisms to enhance the search capability under complex constraints and reduce redundant searches. Results from 30 independent runs show that, with population sizes of 150 and 300, the proposed algorithm achieves mean final fitness values of 15,162.784 and 15,870.237, respectively, corresponding to improvements of 62.54% and 63.84% over the single-point crossover strategy and 28.11% and 27.23% over the multi-point crossover strategy. Under the high-load condition with 250 candidate observation tasks, the total observation profit achieved by the proposed algorithm in the LEO target scenario is 30.09%, 9.14%, 13.78%, and 21.60% higher than that achieved by GA, TGA, ACO, and PSO, respectively. In the mixed MEO and HEO target scenario, the corresponding improvements are 29.94%, 6.26%, 13.22%, and 19.67%, respectively. The results demonstrate that the proposed method provides high-quality scheduling solutions, stable performance, and good adaptability to different orbital scenarios under complex resource constraints and high task loads.

Keywords: satellite observation tasks; optical ground stations; resource scheduling

1. Introduction

As the number of satellites in orbit continues to grow, the imbalance between limited ground resources and increasing demands for satellite observation, tracking, telemetry and command (TT&C), and communication has become increasingly pronounced. Constrained by orbital motion and ground-station locations, satellite-ground tasks can be executed only within limited visibility windows and are subject to resource exclusivity, task switching, and temporal conflict constraints. Consequently, coordinating satellite tasks and ground resources has become an important combinatorial optimization problem in satellite operations management [1]. Existing studies have employed mathematical programming, dynamic programming, and decomposition-based optimization to address the coupling among visibility windows, ground resources, and task rewards, establishing a foundation for resource scheduling across multiple satellites and ground stations [2–7].

Metaheuristic algorithms have been widely adopted as task volumes and resource conflicts increase. Ant colony optimization allocates TT&C resources by exploiting visibility arcs and task conflict relationships [8,9]. Li et al. [10] proposed a station-ID-based grouped



Academic Editor: Kyun Ho Lee

Received: 27 August 2026

Revised: 29 September 2026

Accepted: 29 September 2026

Published: 30 September 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and

conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

encoding scheme that improves genetic search by combining station-specific chromosome segments from different parents. Building on this representation of station-level task structure, the present study orders tasks within each station by their window start times and restricts crossover to a local segment within a single station block. Subsequent studies have incorporated local search, population perturbation, and specialized genetic operators to improve solution quality and search efficiency in complex scheduling problems [11–13].

Regarding hybrid search, Xu et al. [14] proposed a tabu-genetic algorithm incorporating heuristic rules. Their method constructs schedules through task-permutation encoding, execution-duration trimming, and hardware allocation, while using a local tabu list, a global tabu list of historical best solutions, and neighborhood-based mutation to reduce repeated search. The present study draws on this search-memory strategy but directly optimizes binary decisions indicating whether complete observation windows are selected. Task combinations are modified through within-station segment exchange and bit flipping, without permuting tasks or trimming observation windows. For single-pass imaging planning of agile satellites, Wu et al. [15] combined composite encoding, conflicting-target replacement, and midpoint bidirectional selection with tabu-based local search. Their search operates on imaging-target sequences under attitude-maneuver constraints, whereas the present study constructs neighborhoods around ground-station window selection, resource conflicts, and cumulative observation requirements. These distinctions highlight that the suitability of a hybrid algorithm depends not only on the combination of search mechanisms but also on the alignment among decision representation, genetic operators, and constraint handling.

Related research has also developed memetic algorithms, learning-guided multiobjective evolutionary algorithms, and variants of the non-dominated sorting genetic algorithm II (NSGA-II) [16–19], while incorporating parental knowledge, elite archives, task conflict information, and reinforcement learning to guide the search [20–23]. Wang et al. proposed an adaptive variable neighborhood search algorithm incorporating the Metropolis acceptance criterion and a tabu list [24], and a surrogate-assisted variable neighborhood search approach for integrated TT&C and data transmission scheduling [25]. Liu et al. [26] further incorporated frequent pattern mining into an iterated adaptive large neighborhood search algorithm, using inferior and superior structures identified in historical solutions to guide task removal and new-solution construction. These studies offer complementary approaches to satellite resource scheduling through population evolution, local exploitation, and the use of historical search information.

Meanwhile, the scope of research has expanded to integrated imaging–downlink scheduling and distributed observation planning [27–30]. Li et al. [31] proposed the AEM-D3QN framework, which uses graph neural networks to extract task features and incorporates an adaptive exploration mechanism for dynamic satellite mission planning. Zhu et al. [32] combined spatiotemporal task decomposition, accessibility–visibility modeling, and deep reinforcement learning to enable coordinated observation by heterogeneous platforms. These studies extend scheduling capabilities to dynamic environments and settings involving coupled resources, further highlighting the importance of tailoring solution methods to task structures and resource characteristics.

This study addresses resource scheduling for ground-based optical stations, treating each complete satellite–station visibility window as a candidate observation task. Once selected, a task occupies its entire window and must satisfy station switching requirements, mutual exclusion of simultaneous observations of the same target across stations, and minimum cumulative observation durations. Building on station-grouped encoding and hybrid tabu search, we propose a station-structured genetic algorithm driven by dual-level tabu memory to maximize the priority-weighted cumulative effective observation duration. The algorithm combines heuristic initialization, local segment crossover within

station blocks, single-bit-flip neighborhoods, and feasibility screening, while using local and global memories to restrict duplicate candidates. The methodological contribution lies in coordinating station-structured operators with search memory under complete-window selection constraints, rather than in introducing dual-level tabu memory as a standalone innovation. Numerical experiments are conducted to evaluate solution quality, convergence behavior, and computational cost.

The remainder of this paper is organized as follows. Section 2 formulates the resource scheduling model for ground-based optical stations. Section 3 presents the proposed algorithm and its principal operators. Section 4 reports and analyzes the experimental results. Section 5 concludes the paper and discusses directions for future research.

2. Mathematical Formulation of Optical Ground Station Resource Scheduling for Satellite Observation Tasks

We consider the Optical Measurement Resource Scheduling (OMRS) problem. Given a set of high-priority target satellites, a set of optical ground stations, and a set of precomputed satellite–station visibility windows within the planning horizon, the objective of the scheduling problem is to determine the final set of observation tasks to be executed from the candidate observation tasks, subject to the temporal requirements of the observation tasks and the resource constraints of the optical ground stations. To focus on multi-station resource competition and observation task scheduling, the model is formulated under deterministic observation conditions. Specifically, it is assumed that the weather conditions throughout the planning horizon are suitable for optical observations, that all ground-based optical stations remain fully operational, and that the satellite–station visibility windows obtained from orbital visibility analysis are deterministic. Accordingly, uncertainties associated with weather variations, equipment failures, and visibility-window prediction errors are not considered in the current model. Let the set of high-priority target satellites be defined as $S = \{s_i \mid i \in I\}$, $I = \{1, 2, \dots, N_S\}$, where N_S denotes the number of high-priority target satellites. The set of optical ground stations is defined as $G = \{g_k \mid k \in K\}$, $K = \{1, 2, \dots, N_G\}$, where N_G denotes the number of optical ground stations. For notational simplicity, $i \in I$ and $k \in K$ are used throughout this paper to index the high-priority target satellites and optical ground stations, respectively. Accordingly, the i -th high-priority target satellite s_i is hereafter referred to as satellite i , and the k -th optical ground station g_k as station k .

For any high-priority target satellite $i \in I$ and optical ground station $k \in K$, the j -th visibility window between satellite i and station k is denoted by $w_{ikj} = [start_{ikj}, end_{ikj})$, where $start_{ikj}$ and end_{ikj} denote the start and end times of the visibility window, respectively. Let $J_{ik} = \{1, \dots, C_{ik}\}$, where C_{ik} denotes the number of visibility windows between satellite i and optical ground station k . The duration of each visibility window is defined as $d_{ikj} = end_{ikj} - start_{ikj}$. For a given satellite–station pair, the visibility windows are indexed in chronological order according to their occurrence times and satisfy $start_{ik1} < start_{ik2} < \dots < start_{ikC_{ik}}$. Each visibility window corresponds to a candidate observation task. Accordingly, the complete set of candidate observation tasks is defined as $T = \{(i, k, j) \mid i \in I, k \in K, j \in J_{ik}\}$, and the total number of candidate tasks is $N_T = |T| = \sum_{i=1}^{N_S} \sum_{k=1}^{N_G} C_{ik}$. To facilitate the formulation of the conflict constraints, an arbitrary task can be compactly represented as $t = (i_t, k_t, j_t) \in T$, where i_t denotes the satellite associated with task t , k_t denotes the optical ground station assigned to task t , and j_t denotes the index of the corresponding visibility window for the satellite–station pair. The start time, end time, and duration of task t are then defined as $start_t = start_{i_t k_t j_t}$, $end_t = end_{i_t k_t j_t}$, and $d_t = end_{i_t k_t j_t} - start_{i_t k_t j_t}$, respectively. The corresponding task-selection variable is denoted by $x_t \equiv x_{i_t k_t j_t}$.

Let p_i denote the priority of high-priority target satellite i . All optical ground stations are assumed to have identical baseline observation capabilities, and differences in observation quality for the same high-priority target across different stations and visibility windows are not considered. Accordingly, all candidate observation tasks associated with the same high-priority target are assigned the same priority, i.e., $p_{ikj} = p_i > 0$. For a successfully scheduled task t , its scheduling reward depends jointly on the priority p_{i_t} of the corresponding target and the duration d_t of its associated visibility window. Let C_1 , with $C_1 \geq 0$, denote the station turnaround time, defined as the fixed time required before an optical ground station can begin executing the next task after completing the current one. The turnaround time C_1 accounts for equipment slewing, stabilization, target reacquisition, and task preparation. Let C_2 denote the minimum continuous observation duration required for an individual optical observation task. Only candidate tasks satisfying $d_t \geq C_2$, are admitted into the scheduling model. Accordingly, the set of admissible tasks is defined as $T_{adm} = \{t \in T \mid d_t \geq C_2\}$, which contains all candidate tasks that satisfy the basic duration requirement and are therefore eligible for scheduling. For any task $t \notin T_{adm}$, no decision variable is introduced; equivalently, its selection variable is fixed as $x_t = 0$. Furthermore, let $D_i^{\min}$ denote the minimum cumulative observation time required for high-priority target satellite i over the entire planning horizon. It is important to distinguish $D_i^{\min}$ from C_2 . Specifically, C_2 specifies the minimum continuous observation duration for an individual task and is used to exclude visibility windows that are too short to provide an effective observation, whereas $D_i^{\min}$ specifies the minimum cumulative observation time required to ensure that each high-priority target satellite receives a sufficient amount of observation data over the planning horizon. The parameter set D is therefore defined as $D = \{C_1, C_2, D_i^{\min} \mid i \in I\}$.

The objective function is defined in Equation (1) as follows:

$$\max Z = \sum_{t \in T_{adm}} p_{i_t} d_t x_t \quad (1)$$

Subject to the minimum cumulative observation-time requirement for each high-priority target, the model maximizes the total priority-weighted effective observation time, thereby preferentially allocating the limited optical ground-station resources to candidate observation tasks associated with higher-priority targets and longer effective visibility windows. In this way, longer continuous visibility windows can be more effectively utilized, while the total priority-weighted effective observation time is maximized. The binary decision variable x_t in Equation (1) indicates whether candidate observation task t is selected for execution and is defined in Equation (2) as follows:

$$x_t = \begin{cases} 1 & \text{if the } j\text{-th visibility window of satellite } i \text{ at station } k \text{ is selected} \\ 0 & \text{if the } j\text{-th visibility window of satellite } i \text{ at station } k \text{ is not selected} \end{cases} \quad (2)$$

Constraint 1: Because each candidate observation task corresponds to a complete satellite–station visibility window, only tasks whose visibility-window durations reach the minimum effective continuous observation duration C_2 are meaningful for scheduling. Accordingly, the following constraint must be satisfied, as given in Equation (3):

$$end_t - start_t \geq C_2, \forall t \in T_{adm} \quad (3)$$

For any task $\forall t \in T \setminus T_{adm}$, the duration of the corresponding visibility window is insufficient to satisfy the minimum duration requirement for a single effective optical observation. Therefore, such a task is excluded from subsequent scheduling, which is equivalently enforced by setting $x_t = 0$. Accordingly, when $x_t = 1$, task t is selected and

occupies the entire corresponding visibility window; when $x_t = 0$, the task is not selected and the corresponding visibility window yields no observation reward.

Constraint 2: Optical ground stations are exclusive-use observation resources; therefore, a given station can continuously track and observe only one high-priority target at any given time. In addition, after completing one task, the same station requires a fixed turnaround time C_1 before it can begin the next observation task. For any two distinct schedulable tasks t and q , let $\forall\{t, q\} \in C_G$, where the ground-station resource-conflict set C_G is defined in Equation (4) as:

$$C_G = \left\{ \{t, q\} \left| \begin{array}{l} t, q \in T_{adm}, t \neq q, \\ k_t = k_q, \\ end_t + C_1 > start_q, \\ end_q + C_1 > start_t \end{array} \right. \right\} \quad (4)$$

If tasks t and q are assigned to the same optical ground station, that is, $k_t = k_q$, they can both be included in the final schedule only if at least one of the following temporal conditions is satisfied: $end_t + C_1 \leq start_q$ or $end_q + C_1 \leq start_t$. These conditions ensure that one task is completed, together with the required station turnaround time, before the other task begins. Consequently, if $end_t + C_1 > start_q$ and $end_q + C_1 > start_t$ hold simultaneously, tasks t and q constitute a ground-station resource conflict. In this case, the following constraint, given in Equation (5), must be imposed:

$$x_t + x_q \leq 1, \forall\{t, q\} \in C_G \quad (5)$$

This constraint ensures that an optical ground station cannot execute two conflicting observation tasks simultaneously and that a fixed turnaround time of at least C_1 is reserved between two consecutive tasks.

Constraint 3: Simultaneous redundant observation of the same high-priority target satellite by multiple optical ground stations is not considered. Therefore, for any two distinct tasks t and q , let $\forall\{t, q\} \in C_S$, where the same-target observation-conflict set C_S is defined in Equation (6) as:

$$C_S = \left\{ \{t, q\} \left| \begin{array}{l} t, q \in T_{adm}, t \neq q, \\ i_t = i_q, \\ k_t \neq k_q, \\ start_t < end_q, \\ start_q < end_t \end{array} \right. \right\} \quad (6)$$

If tasks t and q correspond to the same high-priority target satellite, that is, $i_t = i_q$, but are assigned to different optical ground stations, that is, $k_t \neq k_q$, and their visibility windows overlap in time, such that $start_t < end_q$ and $start_q < end_t$, the two tasks cannot be selected simultaneously. Accordingly, the following constraint must be satisfied, as given in Equation (7):

$$x_t + x_q \leq 1, \forall\{t, q\} \in C_S \quad (7)$$

This constraint ensures that, at any given time, each high-priority target satellite is observed by at most one optical ground station.

Constraint 4: To ensure that sufficient observation data are available for subsequent orbit analysis and orbital maneuver detection, the cumulative observation time of all successfully scheduled tasks associated with each high-priority target satellite $i \in I$ over the entire planning horizon must be no less than the prescribed minimum cumulative

observation time $D_i^{\min}$. Therefore, the following constraint must be satisfied, as given in Equation (8):

$$D_i^{\min} \leq \sum_{\substack{t \in T_{adm} \\ i_t = t}} d_t x_t, \forall i \in I \quad (8)$$

This constraint ensures that high-priority target i receives at least the prescribed amount of cumulative effective observation data over the entire planning horizon.

3. Station-Structured Genetic Algorithm Driven by Dual-Level Tabu Memory

To address the challenges of a large number of candidate tasks, strongly coupled resource constraints, and redundant search caused by conventional genetic operations in the OMRS problem, this study proposes a station-structured genetic algorithm driven by dual-level tabu memory. The proposed method constructs grouped binary chromosomes using optical ground stations as the basic resource units and integrates heuristic population initialization, fitness-proportionate selection, station-structured crossover and mutation, and a dual-level tabu-memory mechanism consisting of local and global tabu memories. By maintaining scheduling feasibility, these mechanisms improve search efficiency while preserving population diversity. Finally, an elitist strategy is employed for population updating, and the feasible schedule with the highest fitness over the planning horizon is returned as the final solution.

3.1. Chromosome Encoding Based on Optical Ground Station Grouping

This study employs a binary chromosome representation to encode optical ground-station resource schedules. Because each candidate observation task in the admissible task set T_{adm} corresponds to a binary decision variable x_t in the mathematical model, each gene in the chromosome has a one-to-one correspondence with a specific candidate observation task. Specifically, $x_t = 1$ indicates that task t is selected for execution, whereas $x_t = 0$ indicates that the task is not selected. Accordingly, the length of the complete chromosome is equal to the number of admissible candidate observation tasks, as given in Equation (9):

$$\text{length}(X) = |T_{adm}| \quad (9)$$

This encoding scheme directly adopts the binary task-selection variables defined in the mathematical model. A predetermined ordering of the candidate tasks establishes a unique correspondence between each gene position and its associated task, thereby eliminating the need for complex indirect encoding and decoding procedures. Each candidate task t uniquely specifies its associated high-priority target satellite i_t , optical ground station k_t , and visibility window j_t . Moreover, only tasks satisfying the minimum continuous observation-duration requirement are included in T_{adm} . Consequently, the chromosome encodes only those candidate observation tasks that are eligible for scheduling.

Considering that optical ground stations are exclusive-use observation resources in the OMRS problem, schedulable candidate tasks are further grouped according to optical ground-station ID to establish an explicit correspondence between the chromosome structure and the actual station-resource configuration. For any optical ground station $k \in K$, the subset of candidate observation tasks assigned to station k is defined in Equation (10) as:

$$T_k = \{t \in T_{adm} \mid k_t = k\}, k \in K \quad (10)$$

Because each candidate observation task is associated with a unique optical ground station, the task subsets associated with different stations are mutually disjoint, and the union of all station-specific task subsets constitutes the complete admissible task set, as expressed in Equation (11):

$$T_{adm} = \bigcup_{k \in K} T_k, T_{k_1} \cap T_{k_2} = \emptyset, k_1 \neq k_2 \quad (11)$$

For the candidate observation tasks associated with the same optical ground station k , the tasks are sorted in ascending order of their start times $start_t$. When multiple tasks have identical start times, a deterministic ordering is further established according to their high-priority target-satellite indices i_t and visibility-window indices j_t . Let $t_{k,h}$ denote the h -th candidate observation task associated with station k after sorting. Accordingly, the candidate tasks in T_k are ordered by start time and indexed sequentially, as expressed in Equation (12):

$$start_{t_{k,1}} \leq start_{t_{k,2}} \leq \dots \leq start_{t_{k,|T_k|}} \quad (12)$$

Organizing the candidate tasks within each station in chronological order enables the local gene arrangement in the chromosome to reflect the actual temporal sequence of task execution at the corresponding optical ground station over the planning horizon. Accordingly, the chromosome segment associated with optical ground station k is defined in Equation (13) as:

$$X_k = (x_{t_{k,1}}, x_{t_{k,2}}, x_{t_{k,3}}, \dots, x_{t_{k,|T_k|}}) \quad (13)$$

where X_k contains the selection states of all admissible candidate observation tasks assigned to optical ground station k , and $x_{t_{k,h}}$ denotes the binary task-selection variable associated with candidate task $t_{k,h}$. By concatenating the chromosome segments associated with individual stations in ascending order of optical ground-station ID, the complete chromosome is obtained as expressed in Equation (14):

$$X = (X_1, X_2, X_3, \dots, X_k, \dots, X_{N_G}) \quad (14)$$

Therefore, the k -th chromosome segment X_k represents the local observation-task configuration of optical ground station k for the current individual, whereas the complete chromosome X represents the overall observation-resource allocation across all optical ground stations over the planning horizon. This representation maintains a one-to-one correspondence with the task-selection decision variables x_t defined in the mathematical model and provides a unified representation of individuals for the subsequent genetic operations based on candidate-task selection states.

3.2. Heuristic Rule-Based Initial Population Generation

Based on the chromosome encoding scheme described above, a heuristic rule-based strategy is further developed for initial population generation. Since each binary gene in the complete chromosome corresponds uniquely to a candidate observation task in T_{adm} , the generation of an initial individual essentially involves determining the value of the decision variable x_t associated with each candidate task while satisfying all model constraints.

First, candidate observation tasks with relatively short durations are preferentially selected from the admissible task set T_{adm} . A task is incorporated into the current schedule only if the ground-station resource exclusivity and turnaround constraints, as well as the simultaneous same-target observation exclusivity constraint, remain satisfied. This strategy enhances the flexibility of task allocation under limited observation resources. Meanwhile, during the construction of each initial individual, the observation requirements

of all high-priority targets are given priority to ensure that the resulting individual satisfies the minimum cumulative observation-time requirement $D_i^{\min}$.

On this basis, candidate observation tasks whose durations exceed a predefined threshold and that have not yet been selected are subjected to random insertion attempts. Because a fixed-position binary encoding scheme is adopted, task insertion is implemented by changing the corresponding decision variable x_t from 0 to 1, without altering the encoding position of the task within the chromosome. If the task to be inserted does not conflict with any currently selected task, it is retained directly. If a resource conflict occurs, the inserted task is used to replace the selected task or tasks that conflict with it, after which the total priority-weighted effective observation time is recalculated. The replacement is retained only if the modified schedule satisfies all model constraints and yields an improved objective value; otherwise, the original schedule is restored. For each candidate task, at most five random insertion attempts are performed. If no improved feasible schedule can be obtained after these attempts, the task is discarded. By prioritizing shorter-duration tasks to construct high-quality initial solutions while using the random insertion and replacement of longer-duration tasks to introduce differences among individuals, the proposed strategy achieves a balance between initial population quality and population diversity. Since each binary variable in the chromosome corresponds uniquely to a specific candidate observation task and the variables are grouped according to optical ground-station ID, the above definition of task “insertion” remains fully consistent with the chromosome encoding scheme described previously.

3.3. Fitness Evaluation Based on Priority-Weighted Observation Reward

Fitness is used to evaluate the quality of the optical ground-station resource schedule represented by a chromosome and provides a unified criterion for candidate-solution evaluation during subsequent parent selection, offspring competition, and tabu-search procedures. Since the binary chromosome adopted in this study has a one-to-one correspondence with the task-selection decision variables x_t in the mathematical model, the fitness of an individual can be calculated directly from the original optimization objective, without requiring an additional mapping between the chromosome representation and the objective function. Each candidate observation task $t \in T_{adm}$ is uniquely associated with a high-priority target satellite i_t , an optical ground station k_t , and a visibility window, while its inclusion in the schedule is determined by the binary decision variable x_t .

When task t is selected for execution, its effective contribution to the schedule is jointly determined by the target priority and the continuous effective observation duration, namely $p_{i_t} d_t$. Consistent with the objective of the mathematical model, the fitness function is designed to prioritize high-priority targets while fully exploiting longer effective visibility windows, thereby maximizing the total priority-weighted effective observation time over the entire planning horizon. For any chromosome X , the fitness is defined in Equation (15) as follows:

$$Fit(X) = \sum_{t \in T_{adm}} p_{i_t} d_t x_t \quad (15)$$

When $x_t = 1$, task t is included in the current observation schedule and contributes $p_{i_t} d_t$ to the fitness of the individual. When $x_t = 0$, the task is not selected and therefore contributes no observation reward. Accordingly, $Fit(X)$ represents the total priority-weighted effective observation time achieved by the schedule represented by chromosome X over the entire planning horizon. For two feasible individuals X^a and X^b , if the condition given in Equation (16) is satisfied:

$$Fit(X^a) > Fit(X^b) \quad (16)$$

If the above condition is satisfied, the optical ground-station resource schedule represented by X^a is considered superior to that represented by X^b . A larger fitness value indicates that the current schedule achieves a greater total priority-weighted effective observation time over the planning horizon. Specifically, while effectively utilizing the available satellite–station visibility windows, the schedule allocates more valuable observation resources to higher-priority target satellites.

3.4. Parent Selection via Fitness-Proportionate Cumulative Probability Sampling

The selection operator is used to identify parent individuals from the current population for subsequent genetic operations. The associated selection pressure directly affects both the retention and propagation of high-quality schedules and the preservation of population diversity. In this study, a fitness-proportionate roulette-wheel selection strategy is adopted, whereby feasible schedules with higher fitness values are assigned higher probabilities of being selected as parents, while other feasible individuals retain a nonzero probability of participation. This mechanism maintains a balance between preserving high-quality individuals and retaining sufficient stochasticity in the population search process.

Let the current population size be N_{pop} , and denote the m -th chromosome by X^m , where $m = 1, 2, \dots, N_{pop}$. According to the fitness function defined in the preceding subsection, the fitness of the m -th individual is given in Equation (17) as follows:

$$Fit(X^m) = \sum_{t \in T_{adm}} p_{it} d_t x_t^m \quad (17)$$

where $x_t^m \in \{0, 1\}$ denotes the selection state of candidate observation task t in chromosome X^m . Since parent selection in this study is performed only on feasible individuals that satisfy all model constraints, Equation (17) can be directly used to evaluate the relative quality of different schedules. Based on the proportion of an individual's fitness to the total fitness of the current population, the probability that chromosome X^m is selected as a parent is defined in Equation (18) as follows:

$$P_m = \frac{Fit(X^m)}{\sum_{r=1}^{N_{pop}} Fit(X^r)}, m = 1, 2, \dots, N_{pop} \quad (18)$$

where P_m satisfies the normalization condition given in Equation (19):

$$\sum_{m=1}^{N_{pop}} P_m = 1 \quad (19)$$

Therefore, an individual with a higher fitness value occupies a larger probability interval during the selection process and consequently has a higher probability of being selected as a parent. This mechanism provides more reproductive opportunities to schedules with greater total priority-weighted effective observation time, while avoiding the deterministic elimination of other feasible individuals. In this way, sufficient selection pressure is maintained while a certain degree of stochasticity is preserved within the population.

The cumulative selection probability of the first m individuals is further defined in Equation (20) as follows:

$$Q_m = \sum_{r=1}^m P_r, m = 1, 2, \dots, N_{pop} \quad (20)$$

Let $Q_0 = 0$. Since $Q_{N_{pop}} = 1$, the cumulative probability sequence partitions the interval $(0, 1]$ into N_{pop} nonoverlapping probability intervals. The interval associated with the m -th individual is given by $(Q_{m-1}, Q_m]$. During each parent-selection operation, a random

number ε is generated from a uniform distribution over (0,1]. If the condition given in Equation (21) is satisfied:

$$Q_{m-1} < \varepsilon \leq Q_m \quad (21)$$

Chromosome X^m is selected and added to the parent population. This independent sampling procedure is repeated until the parent population reaches the predefined size. Roulette-wheel selection is performed with replacement, allowing the same high-fitness individual to be selected multiple times and thereby enhancing the propagation of high-quality scheduling structures during subsequent genetic operations.

3.5. Station-Structured Crossover and Mutation Strategy Based on Dual-Level Tabu Memory

The crossover operator generates new candidate schedules by recombining task-selection information from the parent chromosomes and therefore plays a key role in expanding the search space of the genetic algorithm. As the evolutionary process proceeds, the similarity among individuals in the population gradually increases, making conventional random crossover more likely to regenerate chromosomes that have already been visited and thereby reducing effective search efficiency. To mitigate this problem, local and global tabu lists are incorporated into the crossover process to exploit historical search information and suppress the repeated generation of previously visited individuals.

Based on the optical-ground-station-grouped binary encoding established in Section 3.1, the two parent chromosomes obtained through the parent-selection procedure described in Section 3.4 are denoted as $X^a = (X_1^a, X_2^a, \dots, X_{N_G}^a)$, $X^b = (X_1^b, X_2^b, \dots, X_{N_G}^b)$, respectively. For optical ground station k , the corresponding chromosome segments in the two parent chromosomes are expressed as $X_k^a = (x_{t_{k,1}}^a, x_{t_{k,2}}^a, \dots, x_{t_{k,|T_k|}}^a)$ and $X_k^b = (x_{t_{k,1}}^b, x_{t_{k,2}}^b, \dots, x_{t_{k,|T_k|}}^b)$, respectively. When crossover is performed according to the prescribed crossover probability, an optical ground station k is randomly selected, and two crossover positions r and s are randomly determined within the corresponding chromosome segments X_k^a and X_k^b , subject to $1 \leq r \leq s \leq |T_k|$. The binary selection states of the candidate observation tasks located within the interval $[r, s]$ are then exchanged between the two parent chromosomes. For any $h = 1, 2, \dots, |T_k|$, the gene states of the two candidate offspring in the chromosome segment corresponding to station k are defined in Equations (22) and (23), respectively:

$$x_{t_{k,h}}^{c1} = \begin{cases} x_{t_{k,h}}^b, & r \leq h \leq s \\ x_{t_{k,h}}^a, & h < r \text{ or } h > s \end{cases} \quad (22)$$

$$x_{t_{k,h}}^{c2} = \begin{cases} x_{t_{k,h}}^a, & r \leq h \leq s \\ x_{t_{k,h}}^b, & h < r \text{ or } h > s \end{cases} \quad (23)$$

This procedure yields two complete candidate offspring, X^{c1} and X^{c2} . If the gene states of the two parent chromosomes are identical within the selected crossover interval, the crossover positions are reselected to avoid generating offspring that are completely identical to their parents and thus constitute ineffective crossover results. Because the proposed method adopts a fixed-position binary encoding scheme, crossover alters only the selection states x_t of candidate observation tasks and does not change the correspondence between candidate tasks and gene positions. Therefore, unlike permutation-based encoding schemes, the proposed representation does not suffer from duplicated or missing task indices and requires no additional mapping or repair procedure.

To reduce the generation of duplicate candidate solutions, a local tabu set L_{tabu} and a global tabu set G_{tabu} are introduced. Both tabu sets are implemented using hash sets to store complete binary chromosomes, and duplicate candidate solutions are identified by directly comparing the complete chromosomes. The local tabu set is initialized as empty

at the beginning of each crossover or mutation operation and records only the candidate chromosomes visited during the current operation. The maximum number of attempts for each crossover and mutation operation is set to $MaxTry = 20$. Since each crossover operation generates at most two candidate offspring, the local tabu set used during crossover stores at most 40 candidate chromosomes. Similarly, because each mutation attempt generates at most 10 neighborhood candidate solutions, the local tabu set used during mutation stores at most 200 candidate chromosomes. Candidate solutions that are contained in L_{tabu} , G_{tabu} , or the current offspring set are rejected. After the current crossover or mutation operation is completed, L_{tabu} is cleared.

After the initial population is generated, the current best chromosome is inserted into G_{tabu} . Thereafter, following each population update, the best feasible chromosome of the current generation is added to G_{tabu} , while chromosomes already stored in the set are not inserted repeatedly. The global tabu set remains active throughout a single independent run and is reinitialized at the beginning of the next run. Therefore, $|G_{tabu}| \leq MaxGen + 1$. When $MaxGen = 500$ the global tabu set contains at most 501 chromosomes. If no acceptable candidate offspring is generated after the maximum number of attempts is reached, the current crossover or mutation operation is terminated.

Tabu checking is performed by comparing the task-selection states of complete binary chromosomes. For the two candidate offspring, the following conditions must be satisfied independently: $X^{c1} \notin L_{tabu}, X^{c1} \notin G_{tabu}, X^{c2} \notin L_{tabu}, X^{c2} \notin G_{tabu}$. In addition, neither candidate offspring is allowed to duplicate any offspring already generated in the current crossover process. The crossover result is accepted only when both X^{c1} and X^{c2} satisfy all of the above conditions. If either offspring is found in the local tabu list, the global tabu list, or the set of offspring already generated during the current crossover process, the crossover result is rejected, and new crossover positions are selected to generate another pair of candidate offspring. The detailed procedure of the crossover algorithm is presented in Algorithm 1.

Algorithm 1: Dual-Tabu-Memory Station-Structured Crossover Operator

Input: Parent chromosomes X^a and X^b ; station-task groups $\{T_k\}$; global tabu set G_{tabu} ; previously generated crossover offspring in the current generation; maximum attempts $MaxTry = 20$.

Output: Accepted offspring X^{c1} and X^{c2} , or $\emptyset$ if no acceptable offspring is generated.

```

1: Initialize  $L_{tabu} \leftarrow \emptyset$ .
2: if no  $k \in K$  satisfies  $|T_k| \geq 2$  and  $X_k^a \neq X_k^b$  then
3:   return  $\emptyset$ .
4: end if
5: for at most  $MaxTry$  attempts do
6:   Randomly select  $k \in K$  satisfying  $|T_k| \geq 2$  and  $X_k^a \neq X_k^b$ .
7:   Randomly select  $r$  and  $s$  satisfying  $1 \leq r < s \leq |T_k|$ .
8:   if the gene states of  $X_k^a$  and  $X_k^b$  are identical over  $[r, s]$  then
9:     continue.
10:  end if
11:  Generate  $X^{c1}$  and  $X^{c2}$  by exchanging the gene states of  $X_k^a$  and  $X_k^b$  over  $[r, s]$ .
12:  if  $X^{c1} \notin (L_{tabu} \cup G_{tabu})$ ,  $(X^{c2} \notin L_{tabu} \cup G_{tabu})$ , and neither offspring duplicates
any previously generated crossover offspring then
13:    return  $X^{c1}, X^{c2}$ .
14:  end if
15:   $L_{tabu} \leftarrow L_{tabu} \cup \{X^{c1}, X^{c2}\}$ .
16: end for
17: return  $\emptyset$ .
```

The subsequent mutation operation enhances population diversity and local search capability by modifying local task-selection states. Since a fixed-position binary encoding scheme is adopted in this study, a single-gene flip is employed as the basic mutation operator and is combined with neighborhood search and the dual-level tabu-memory mechanism to construct the mutation strategy. For a chromosome X to be mutated, an optical ground station $k \in K$ is first selected at random, and a gene is then randomly chosen from its corresponding candidate-task set T_k . According to the task ordering defined previously, the selected gene corresponds to candidate task $t_{k,h}$, where $1 \leq h \leq |T_k|$, $t_{k,h} \in T_k$. The binary selection state of this task is flipped according to $x_{t_{k,h}}^{mut} = 1 - x_{t_{k,h}}$, while all other gene states remain unchanged. When $x_{k,h} = 0$, the flip indicates that the candidate task is added to the current schedule; when $x_{k,h} = 1$, the task is removed from the schedule. To reduce the randomness associated with a single stochastic bit flip, multiple single-bit flips are performed at different gene positions within the selected station segment X_k , using the original chromosome as the reference solution, thereby generating a set of neighboring candidate solutions. Since a bit flip may alter the feasibility of the original schedule, each generated candidate solution is checked against the ground-station resource-conflict constraint C_G , the same-target observation-conflict constraint C_S , and the minimum cumulative observation-time requirement $D_i^{\min}$. Only candidate solutions satisfying all scheduling constraints are retained for fitness evaluation, whereas infeasible candidates are discarded directly. The feasible neighboring solutions are then evaluated using the fitness function, and the individual with the highest fitness is selected. This solution is further compared with the feasible individual obtained from the initial random mutation, and the better of the two is retained as the current candidate mutation solution.

Let the final candidate mutation chromosome be denoted by X^{best} . The local tabu list L_{tabu} and the global tabu list G_{tabu} introduced above are retained in the mutation procedure. The local tabu list L_{tabu} is reinitialized at the beginning of each new local mutation search and is used to record candidate chromosomes that have already been visited during the current mutation process. The global tabu list G_{tabu} , which is shared with the crossover operator, stores the historically best chromosomes identified during the evolutionary process. The current mutation result is accepted only if $X^{best} \notin L_{tabu}$ and $X^{best} \notin G_{tabu}$, and the chromosome does not duplicate any mutation individual that has already been generated in the current mutation process. If X^{best} matches a tabu record, it is added to L_{tabu} , and a new set of neighboring candidate solutions is subsequently generated. The detailed procedure of the mutation operation is presented in Algorithm 2.

After the crossover and mutation operations, the generated candidate offspring are subjected to a feasibility check, and only those satisfying all scheduling constraints are retained. The feasible offspring are then combined with the current population for fitness evaluation, and the top N_{pop} individuals ranked in descending order of fitness are selected to form the next-generation population. Since the best individual from the current population always participates in the survivor selection process, the best fitness value of the next generation cannot be lower than that of the current generation, provided that the scheduling instance, objective function, and constraints remain unchanged. Therefore, this elitist retention strategy guarantees a non-decreasing best fitness value across generations, while still allowing crossover and mutation to generate candidate solutions with relatively lower fitness values, thereby facilitating the exploration of alternative task combinations.

After each population update, the chromosome with the highest fitness value in the current population is added to the global tabu set G_{tabu} . The local tabu set L_{tabu} is used to prevent repeated searches within a single crossover or mutation operation, whereas G_{tabu} stores the historically best individuals obtained during the evolutionary process, thereby reducing redundant searches across generations and improving search efficiency.

Algorithm 2: Dual-Tabu-Memory Station-Structured Mutation Operator

Input: Chromosome X ; station-task groups $\{T_k\}$; global tabu set G_{tabu} ; current mutation offspring; maximum attempts $MaxTry = 20$.

Output: Accepted mutant chromosome X^{best} , or the original chromosome X .

```

1: Initialize  $L_{tabu} \leftarrow \emptyset$ ,  $Try \leftarrow 0$ .
2: while  $Try < MaxTry$  do
3:   Randomly select  $k \in K$  and  $h \in \{1, 2, \dots, |T_k|\}$ .
4:    $x_{t_{k,h}}^{mut} \leftarrow 1 - x_{t_{k,h}}$ ; generate single-gene-flip candidates within  $X_k$ .
5:   Retain candidates satisfying  $C_G$ ,  $C_S$ , and  $D_i^{min}$ .
6:   if no feasible candidate remains then  $Try \leftarrow Try + 1$ ; continue.
7:   Select the feasible candidate with the largest fitness as  $X^{best}$ .
8:   if  $X^{best} \notin L_{tabu}$ ,  $X^{best} \notin G_{tabu}$ , and  $X^{best}$  is nonduplicate in current offspring then
9:     return  $X^{best}$ .
10:  end if
11:   $L_{tabu} \leftarrow L_{tabu} \cup \{X^{best}\}$ ;  $Try \leftarrow Try + 1$ .
12: end while
13: return  $X$ .
```

In terms of space complexity, each chromosome consists of $|T_{adm}|$ binary genes, and therefore the storage complexity of a single chromosome is $O(|T_{adm}|)$. According to the tabu-set capacity settings described above, the local tabu sets used during crossover and mutation store at most 40 and 200 candidate chromosomes, respectively, while the global tabu set stores at most $MaxGen + 1$ historically best chromosomes during a single independent run. Consequently, the additional space complexity introduced by the dual-tabu-memory mechanism is $O((200 + MaxGen + 1)|T_{adm}|)$. In addition to the tabu memory, N_{pop} chromosomes must be maintained simultaneously in the population. Therefore, when the complete pairwise task-conflict matrix is not explicitly stored, the dominant space complexity of the proposed algorithm is $O((N_{pop} + MaxGen)|T_{adm}|)$. Since both the capacity of the local tabu set and the number of neighborhood candidate solutions generated in a single attempt are bounded by fixed constants, they do not affect the above asymptotic space complexity. If a complete pairwise conflict matrix is additionally precomputed and stored, an extra $O(|T_{adm}|^2)$ storage space is required. In terms of time complexity, the fitness evaluation of a single chromosome requires traversing all task-selection states and therefore has a complexity of $O(|T_{adm}|)$. Each candidate solution must also be checked against scheduling constraints, including station-resource conflicts, simultaneous duplicate observations of the same target, and minimum cumulative observation-time requirements. If task conflicts are examined in a pairwise manner, the feasibility check for a single candidate solution has a worst-case complexity of $O(|T_{adm}|^2)$. The crossover and mutation procedures additionally involve chromosome construction, tabu checking, and neighborhood candidate screening. For a complete chromosome of length $|T_{adm}|$, chromosome construction and hash computation both require $O(|T_{adm}|)$ time, and a complete chromosome comparison has the same time complexity. Therefore, compared with the $O(|T_{adm}|^2)$ feasibility check, these operations do not change the asymptotic complexity of processing a single candidate solution.

Assuming that the number of candidate offspring generated in each generation is of the same order as N_{pop} , the overall complexity of feasibility checking within one generation is $O(N_{pop}|T_{adm}|^2)$. After merging the current population with the feasible offspring, sorting the individuals according to fitness requires $O(N_{pop} \log N_{pop})$ time. Therefore, over $MaxGen$ generations, the worst-case time complexity of the proposed algorithm is $O(MaxGen(N_{pop}|T_{adm}|^2 + N_{pop} \log N_{pop}))$. The maximum number of generations, $Max-$

Gen, is adopted as the termination criterion. Once the number of generations reaches *MaxGen*, the search process terminates, and the feasible chromosome with the highest fitness value in the final population is returned as the resource scheduling solution for the ground-based optical stations.

4. Results and Discussion

To evaluate the optimization performance of the proposed algorithm for the optical ground-station resource scheduling problem, numerical experiments are conducted under a consistent experimental scenario and computational environment. All algorithms are implemented in Python 3.10 and executed on a computing platform equipped with an AMD Ryzen 9 processor (2.50 GHz) and 16 GB of RAM. All experiments use identical task inputs, constraint settings, and termination criteria. Multiple independent runs are performed to comprehensively assess solution quality, convergence performance, and computational cost, thereby mitigating the influence of stochastic search behavior on the experimental results.

Table 1 summarizes the main parameters of the experimental scenario. Based on these parameters, a simulation scenario comprising high-priority targets and optical ground stations is established using STK 12, and the corresponding satellite–station visibility windows over the planning horizon are generated. The visibility windows are then screened according to the minimum continuous observation-duration requirement for an individual task, yielding the admissible candidate task set used in the subsequent optimization. The priorities and minimum cumulative observation requirements of all high-priority targets are kept identical across the experiments with different algorithms to ensure consistent comparison conditions. The crossover probability, mutation probability, and tabu-memory parameters jointly affect the balance among global exploration, local exploitation, and computational cost. A relatively high crossover probability facilitates information exchange among individuals and expands the search range of task combinations, whereas an excessively high crossover probability may disrupt well-formed high-quality scheduling structures. In contrast, a relatively low crossover probability may weaken information recombination among individuals. The mutation probability is mainly used to maintain population diversity and prevent search stagnation. An excessively low mutation probability may lead to premature population convergence, whereas an excessively high value may introduce excessive random perturbations and weaken the inheritance of high-quality structures. For the tabu-memory mechanism, *MaxTry* determines the maximum number of retry attempts after a candidate solution is rejected, while the number of mutation candidates per attempt determines the width of the local neighborhood search. Appropriately increasing these two parameters can improve the probability of obtaining non-redundant feasible solutions, but may also increase the computational overhead associated with feasibility checking, tabu checking, and fitness evaluation. Therefore, these parameters should be properly balanced in terms of search adequacy, solution stability, and computational efficiency.

Table 1. Experimental Scenario Parameters for Optical Ground-Station Resource Scheduling.

Parameter	Value	Unit
Number of optical ground stations	12	-
Planning horizon	1440	min
Station turnaround time	5	min
Minimum continuous observation duration	30	s

Considering the stochastic search characteristics of genetic algorithms, each experimental configuration is independently executed 30 times. Performance metrics, including

average fitness, convergence behavior, and CPU runtime, are recorded for comprehensive performance evaluation. Under the same experimental scenario, the control parameters of the algorithms are specified in Table 2.

Table 2. Main Parameter Settings of the Genetic Algorithms.

Parameter	Value
Population size	150,300
<i>MaxGen</i>	500
<i>MaxTry</i>	20
Crossover probability	0.7
Mutation probability	0.1

Based on the aforementioned experimental settings, further comparative experiments were conducted to investigate the performance of different genetic strategies and population sizes. The algorithm performance was evaluated from multiple perspectives, including solution quality, convergence characteristics, and computational efficiency. First, the optimization performance of single-point crossover and multi-point crossover genetic algorithms was investigated, followed by the validation of the proposed algorithm under identical experimental conditions.

Under the same task data, constraint conditions, and termination criteria, experiments were conducted using the single-point crossover, multi-point crossover, and the proposed station-structured genetic strategy, respectively. Each strategy was independently executed 30 times with population sizes of 150 and 300. The fitness evolution process, final solution quality, and operational stability were compared to evaluate the performance of different genetic strategies. Figures 1 and 2 illustrate the fitness evolution results of different strategies, where the light-colored curves represent individual runs and the shaded areas indicate the ± 1 standard deviation intervals around the average fitness values.

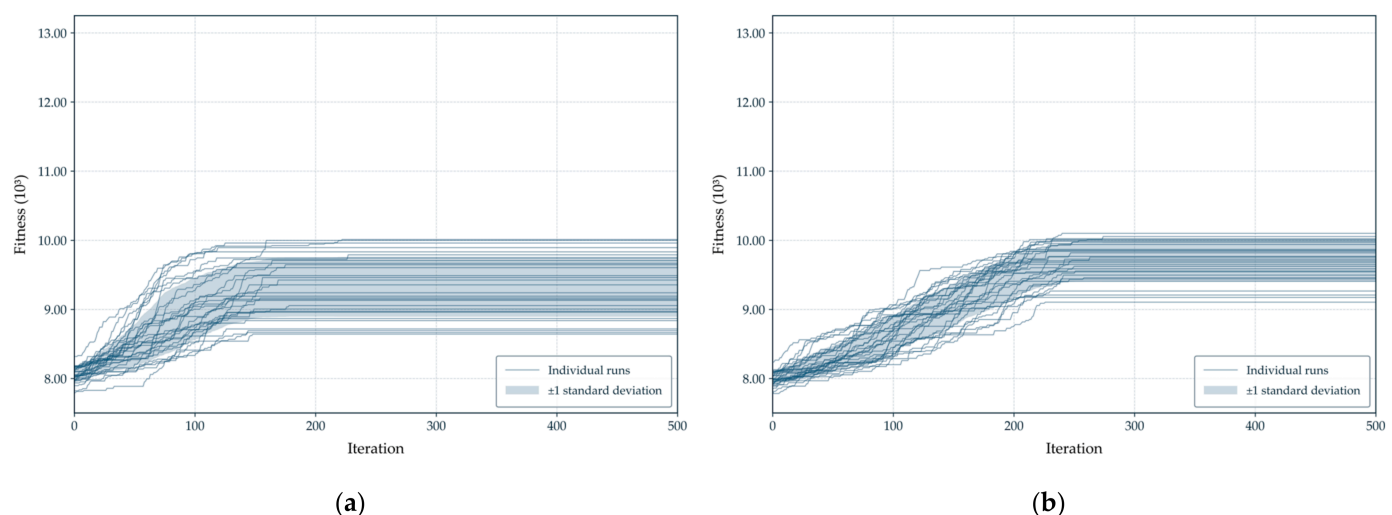


Figure 1. Fitness evolution of the single-point crossover genetic algorithm. (a) Population size of 150; (b) Population size of 300.

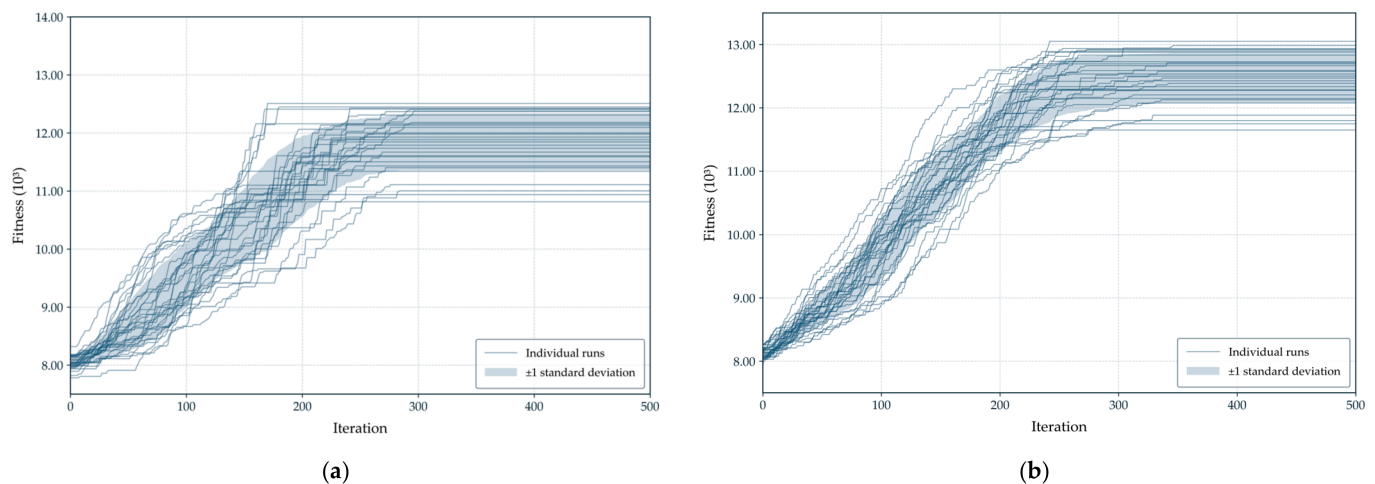


Figure 2. Fitness evolution of the multi-point crossover genetic algorithm. (a) Population size of 150; (b) Population size of 300.

When the population size was set to 150, the average final fitness values of the single-point crossover and multi-point crossover strategies were 9328.417 ± 434.352 and $11,835.732 \pm 491.574$, respectively. Compared with the single-point crossover strategy, the multi-point crossover strategy achieved a 26.88% improvement in the final fitness value, indicating that the recombination of multiple gene segments can generate more diverse task combinations and enhance the exploration capability in high-quality scheduling regions. However, the larger standard deviation observed for the multi-point crossover strategy suggests that although it improves solution quality, it also introduces greater variability among independent runs.

When the population size was increased to 300, the average final fitness values of the single-point crossover and multi-point crossover strategies increased to 9686.253 ± 278.931 and $12,473.586 \pm 392.855$, respectively, corresponding to improvements of 3.84% and 5.39% compared with those obtained with a population size of 150. Meanwhile, the final standard deviations of the two strategies decreased by approximately 35.78% and 20.08%, respectively. These results demonstrate that increasing the population size can enhance population diversity, improve solution quality, and reduce the dispersion of independent runs. Nevertheless, a larger population size also increases the search effort required to achieve a stable convergence state.

Under the same experimental conditions, the proposed dual-tabu-memory-driven station-structured genetic algorithm was independently run 30 times, and the fitness evolution results obtained from these experiments are presented in Figure 3.

Under the same experimental conditions, the proposed dual-tabu-memory-driven station-structured genetic algorithm was independently run 30 times with population sizes of 150 and 300, and the corresponding fitness evolution results are presented in Figure 3. For both population sizes, the fitness curves exhibit a rapid increase during the early stage of the search and gradually stabilize in the later iterations, indicating that the proposed algorithm can quickly obtain high-quality scheduling solutions at the beginning of the search and further improve the solution quality through subsequent local refinement.

When the population size was set to 150, the average final fitness value was $15,162.784 \pm 185.168$. When the population size was increased to 300, the average final fitness value increased to $15,870.237 \pm 176.063$, representing an improvement of 4.67% over that obtained with a population size of 150. As shown in Figure 3a, with a population size of 150, the fitness value increases rapidly during the early iterations and gradually stabilizes after approximately 300 generations. Figure 3b shows that, with a population size of 300,

the fitness value remains nearly unchanged after approximately 350 generations, indicating that a larger population size enables more extensive exploration of the solution space, although a slightly longer evolutionary process is required to achieve stable convergence.

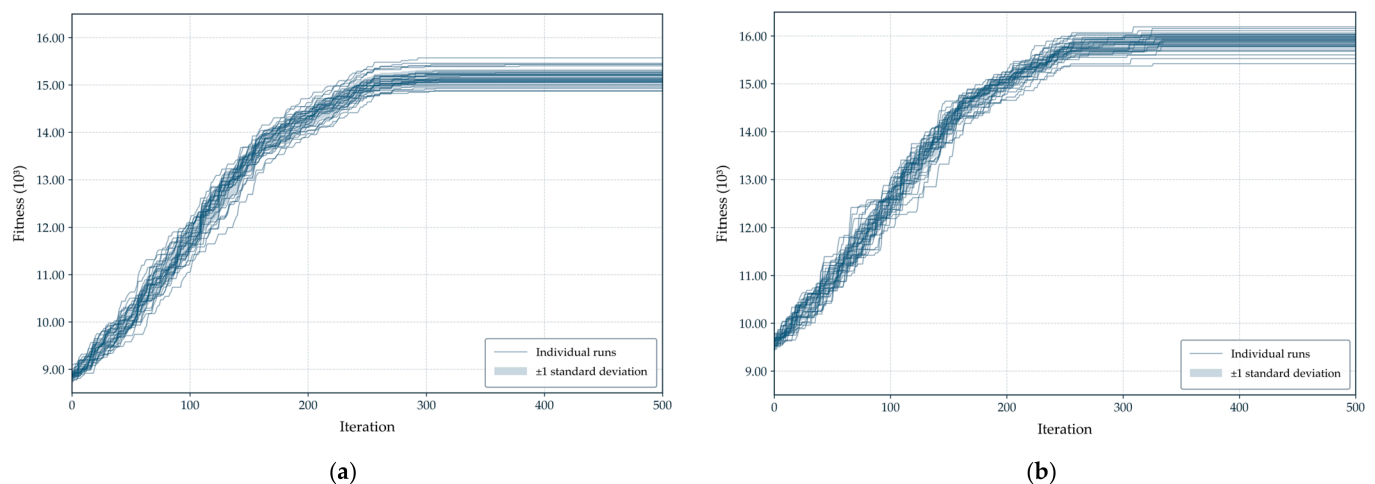


Figure 3. Fitness evolution of the proposed algorithm. (a) Population size of 150; (b) Population size of 300.

Compared with the aforementioned single-point crossover and multi-point crossover strategies, the proposed algorithm improves the average final fitness by 62.54% and 28.11%, respectively, when the population size is 150. For a population size of 300, the corresponding improvements are 63.84% and 27.23%, respectively. In addition, the final fitness standard deviation of the proposed algorithm is lower than those of the two comparison strategies, indicating that the proposed method not only improves scheduling solution quality but also reduces performance fluctuations across independent runs. The Wilcoxon test was further applied to the final fitness values obtained from the 30 independent runs to assess statistical significance. For both population sizes of 150 and 300, the differences in final fitness between the proposed algorithm and each of the two comparison strategies, namely the single-point crossover and multi-point crossover strategies, were statistically significant, with all p -values less than 0.05. Overall, the combined use of station-group-based encoding, structured crossover, neighborhood-enhanced mutation, and dual-tabu-memory mechanisms enhances the utilization of station-resource structures and improves both solution quality and operational stability in complex scheduling spaces.

To further evaluate the performance of different scheduling algorithms under varying task loads in the low-Earth-orbit (LEO) target observation scenario, test instances with different numbers of candidate observation tasks were constructed under a unified ground-station configuration, planning horizon, and set of scheduling constraints. All observation objects were LEO targets. STK 12 was used to establish the orbital and visibility simulation models between the LEO targets and the ground-based optical stations, from which satellite-station visibility windows within the planning horizon were generated. The candidate visibility windows were then screened according to the minimum continuous observation duration requirement. With the number of ground-based optical stations fixed at 12, the planning horizon at 24 h, the inter-task switching time at 5 min, and the minimum continuous observation duration per task at 30 s, the number of candidate observation tasks for LEO targets was set to 50, 100, 150, 200, and 250, respectively, to progressively increase task competition and scheduling conflicts under limited ground-based optical station resources.

For the proposed algorithm, the population size was fixed at 150, the maximum number of generations was set to 500, and the crossover and mutation probabilities were set to 0.7 and 0.1, respectively. The standard genetic algorithm (GA), tabu genetic algorithm (TGA), particle swarm optimization (PSO), and ant colony optimization (ACO) were selected as benchmark methods. These algorithms were used as representative baselines to evaluate the relative performance of the proposed algorithm from the perspective of different search mechanisms. Specifically, GA and TGA represent conventional genetic search and tabu-enhanced genetic search, respectively, whereas PSO and ACO provide comparisons against classical swarm-intelligence optimization methods. It should be noted that these algorithms are intended only as representative baselines for comparisons across different search mechanisms and do not cover all related optimization algorithms discussed in the Introduction. For each task scale, all algorithms were evaluated using the same task inputs, objective function, and constraint set, and each algorithm was independently executed 30 times to reduce the influence of stochastic search behavior on the experimental results. Algorithm performance was evaluated from two perspectives: task coverage capability and scheduling profit. Specifically, the task completion rate was defined as the ratio of the number of successfully scheduled tasks to the total number of candidate tasks, while the total observation profit was defined as the sum, over all completed tasks, of the product of effective observation duration and the priority of the corresponding target. The mean and standard deviation obtained from the 30 independent runs were finally used for statistical analysis, providing a comprehensive evaluation of the solution quality and run-to-run stability of the different algorithms as the LEO task scale increased.

As shown in Figure 4, the total observation profit obtained by all algorithms increases as the number of LEO observation tasks grows from 50 to 250, while the performance gap among the algorithms gradually widens. When the task scale is 50, all five algorithms achieve the same observation profit of 5738 with a standard deviation of 0, indicating that the available ground-station resources are sufficient under low-load conditions and that all algorithms are able to obtain an identical scheduling solution. As the number of tasks increases, competition for limited resources and conflicts among visibility windows become more pronounced, and the advantage of the proposed algorithm gradually emerges. At a task scale of 250, the proposed algorithm achieves an average total observation profit of 20,342.449, representing improvements of 30.09%, 9.14%, 13.78%, and 21.60% over GA, TGA, ACO, and PSO, respectively. Its standard deviation is only 323.084, indicating that the proposed algorithm not only achieves a higher observation profit under complex task conditions, but also maintains good solution stability.

As shown in Figure 5, the task completion rates of all algorithms decrease as the task scale increases. This trend indicates that, with a fixed number of ground stations and an unchanged planning horizon, a heavier task load further intensifies resource competition and visibility-window conflicts. When the task scale is 200, the average task completion rates of GA, TGA, ACO, and PSO are 84.78%, 91.63%, 89.47%, and 86.82%, respectively, whereas the proposed algorithm achieves 92.78%. This corresponds to relative improvements of 9.44%, 1.26%, 3.70%, and 6.86% over the four benchmark algorithms, respectively. Meanwhile, the standard deviation of the task completion rate for the proposed algorithm is only 0.64%, demonstrating that it can maintain both high task coverage and good solution stability under relatively heavy task loads.

Taken together, Figures 4 and 5 show that task completion rate and total observation profit are related, but not in a simple linear manner. Task completion rate mainly reflects the coverage of a scheduling solution, whereas total observation profit also depends on the priorities of the scheduled targets and their effective observation durations. As the number of candidate tasks increases, the proposed algorithm maintains a relatively

high task completion rate and total observation profit. These results suggest that the station-grouped encoding scheme, structured crossover operator, neighborhood-enhanced mutation strategy, and dual-tabu-memory mechanism help identify high-value task combinations and improve solution quality and stability under the tested LEO workloads of 50 to 250 candidate tasks.

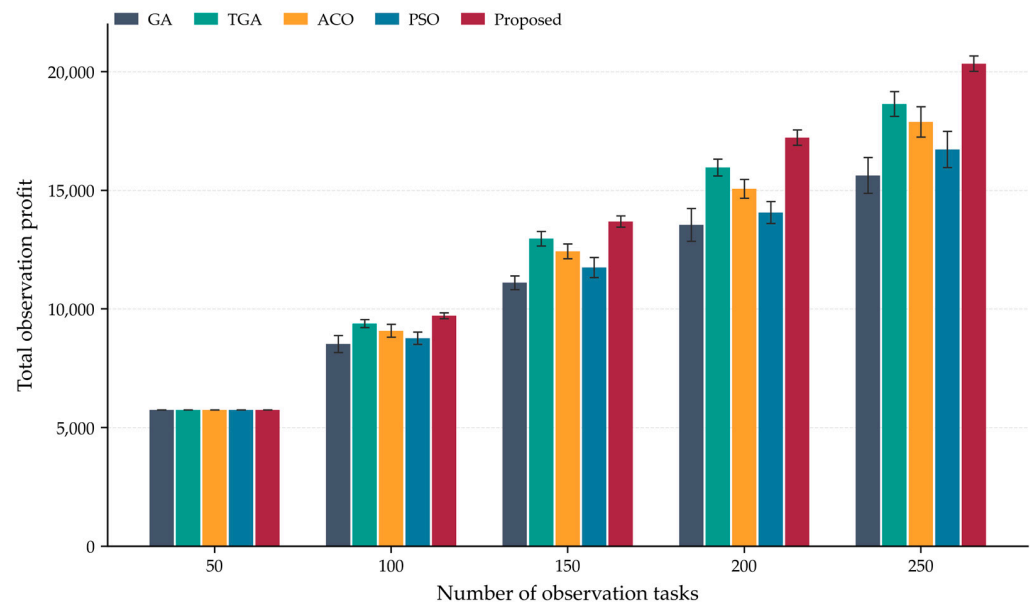


Figure 4. Comparison of the mean total observation profit and standard deviation of different algorithms under different LEO task scales.

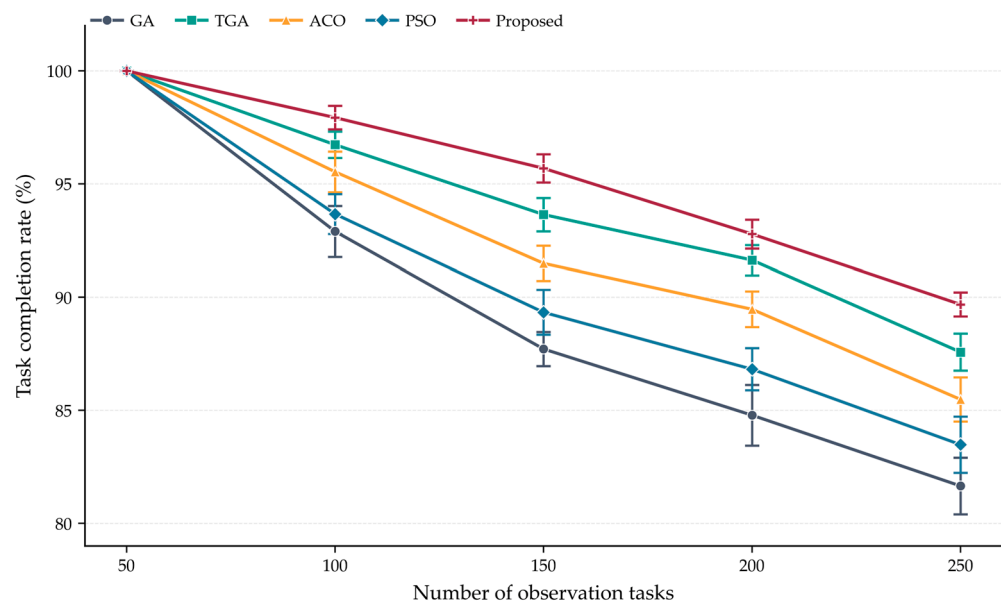


Figure 5. Comparison of the mean task completion rate and standard deviation of different algorithms under different LEO task scales.

To further evaluate the adaptability of the proposed algorithm to targets in different orbital regimes and to more complex visibility-window conditions, a mixed medium-Earth-orbit (MEO) and high-Earth-orbit (HEO) observation scenario is constructed on the basis of the LEO experiments, while keeping the number of ground-based optical stations, the planning horizon, and the main scheduling constraints unchanged. Compared with LEO targets, MEO and HEO targets generally exhibit longer visibility windows, resulting in more complex resource-occupancy characteristics and task-conflict relationships. This scenario

therefore provides a more challenging test for assessing the scheduling performance, solution quality, and stability of the different algorithms under varying task scales.

As shown in Figure 6, the total observation profit obtained by all algorithms increases as the number of mixed MEO–HEO observation tasks grows from 50 to 250, while the performance gap among the algorithms gradually widens. When the task scale is relatively small, the observation profits achieved by the different algorithms are close to one another. As the number of tasks increases, competition for ground-station resources and conflicts among visibility windows become increasingly pronounced, and the advantage of the proposed algorithm gradually emerges. At a task scale of 250, the proposed algorithm achieves a higher average total observation profit than the benchmark algorithms, with improvements of 29.94%, 6.26%, 13.22%, and 19.67% over GA, TGA, ACO, and PSO, respectively. Its standard deviation is only 443.762, which is lower than those of all benchmark algorithms. These results indicate that the proposed algorithm can achieve a higher observation profit under heavy task loads while maintaining good solution stability.

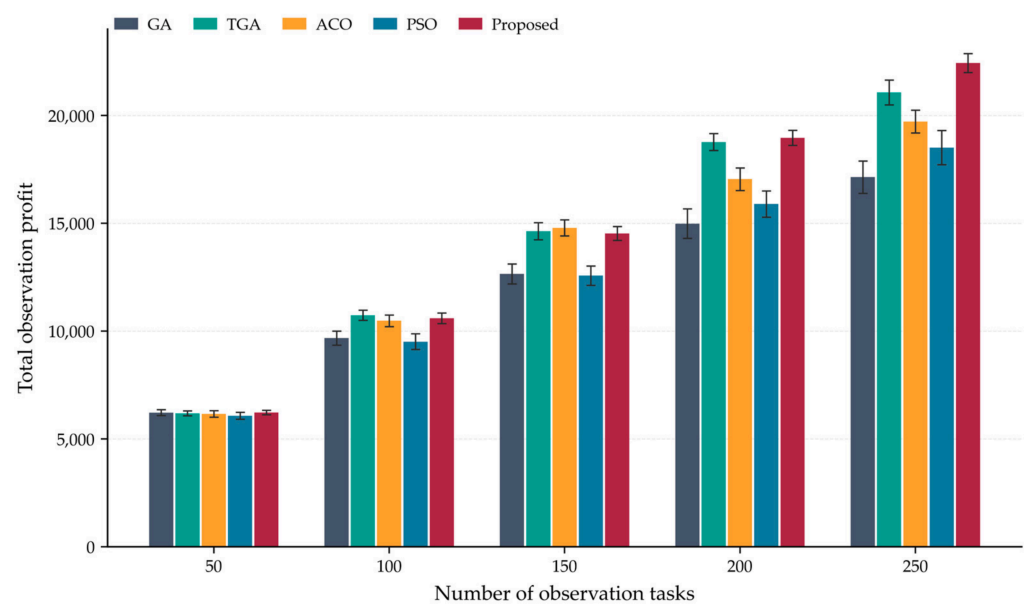


Figure 6. Comparison of the mean total observation profit and standard deviation of different algorithms under different MEO–HEO task scales.

As shown in Figure 7, the task completion rates of all algorithms decrease as the task scale increases. This trend reflects the progressively intensified resource occupancy and task conflicts caused by the longer observation windows of MEO and HEO targets under a fixed ground-station configuration. When the task scale reaches 250, the average task completion rates of GA, TGA, ACO, and PSO are 76.32%, 83.12%, 80.81%, and 78.13%, respectively, whereas the proposed algorithm still achieves 85.61%, with a standard deviation of only 0.82%. Taken together, these two performance metrics demonstrate that the proposed algorithm can effectively balance task coverage and observation profit in the mixed MEO–HEO target scenario, while maintaining high solution quality and good robustness under complex resource-conflict conditions.

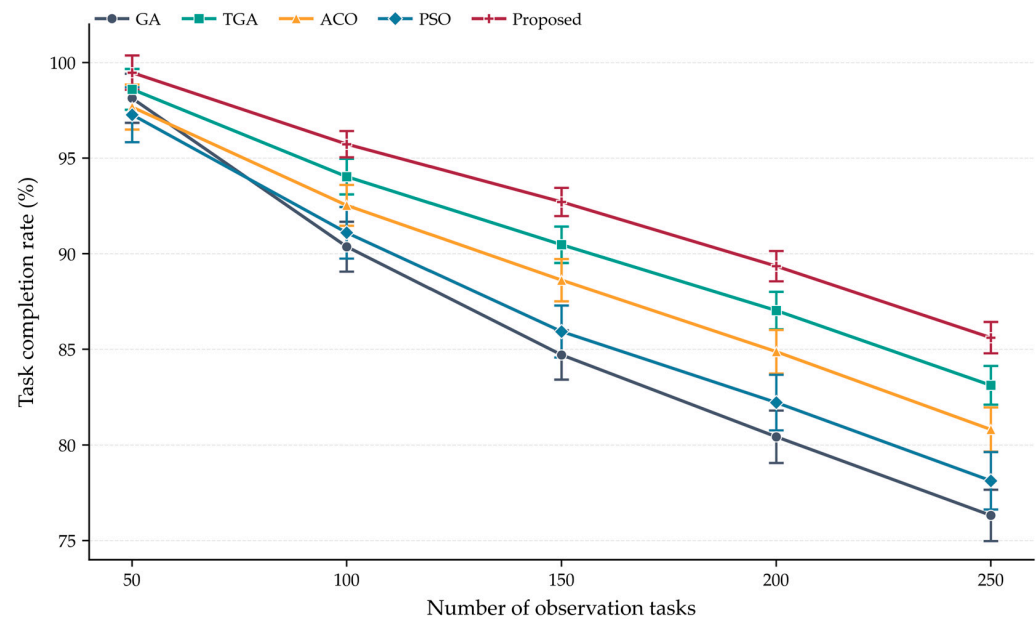


Figure 7. Comparison of the mean task completion rate and standard deviation of different algorithms under different MEO–HEO task scales.

A joint examination of Figures 6 and 7 further shows that, as the scale of the mixed MEO–HEO task set increases, the task completion rates of all algorithms gradually decline, whereas the total observation profit continues to increase. This indicates that intensified task competition also enlarges the selection space for high-value task combinations. Across different task scales, the proposed algorithm consistently maintains both a relatively high task completion rate and a high total observation profit, with comparatively small fluctuations among independent runs. These results demonstrate that the proposed method can effectively balance task coverage capability and observation value under complex visibility-window conflicts, thereby exhibiting strong overall scheduling performance and solution stability.

To further evaluate the computational efficiency and run-to-run stability of the different scheduling algorithms under varying task scales, a comparative runtime experiment was conducted under the same hardware environment and parameter settings as those used in the preceding experiments. Five test instances containing 50, 100, 150, 200, and 250 candidate observation tasks, respectively, were considered, and GA, TGA, ACO, PSO, and the proposed algorithm were compared. To ensure a fair comparison, all algorithms were evaluated using identical task inputs, objective functions, constraint conditions, and termination criteria. In addition, all implementations were developed within a unified Python-based computational framework, with common modules used for data processing, fitness evaluation, and constraint checking. The runtime measurement covered only the computational process from algorithm initialization through iterative search to the generation of the final scheduling solution. Common preprocessing and postprocessing operations, including visibility-window generation, task-data loading, and result-file storage, were excluded from the timing statistics. For each task scale, each algorithm was independently executed 30 times. The mean runtime and standard deviation were then calculated to assess the variation in computational cost and execution stability as the task scale increased.

As shown in Figure 8, the average running time of all algorithms increases as the number of LEO observation tasks grows from 50 to 250, and the differences in computational overhead among the algorithms become more pronounced as the number of tasks increases. With 250 candidate tasks, the proposed algorithm has an average running time

of 894.173 s, which is 7.43% and 24.88% lower than those of TGA and ACO, respectively. Although it runs slightly longer than GA and PSO, their running times remain within a comparable range. The standard deviation of the proposed algorithm's running time is 25.112 s, indicating relatively stable computational overhead across independent runs. Together with the preceding scheduling results, these findings show that the proposed algorithm achieves higher total observation profit and task completion rate with limited additional computation. The station-structured search, neighborhood enhancement, and dual-tabu mechanisms thus help identify high-value task combinations without introducing excessive computational overhead. Overall, the proposed algorithm achieves a favorable balance between solution quality and computational efficiency in the tested LEO scheduling instances of 50 to 250 candidate tasks.

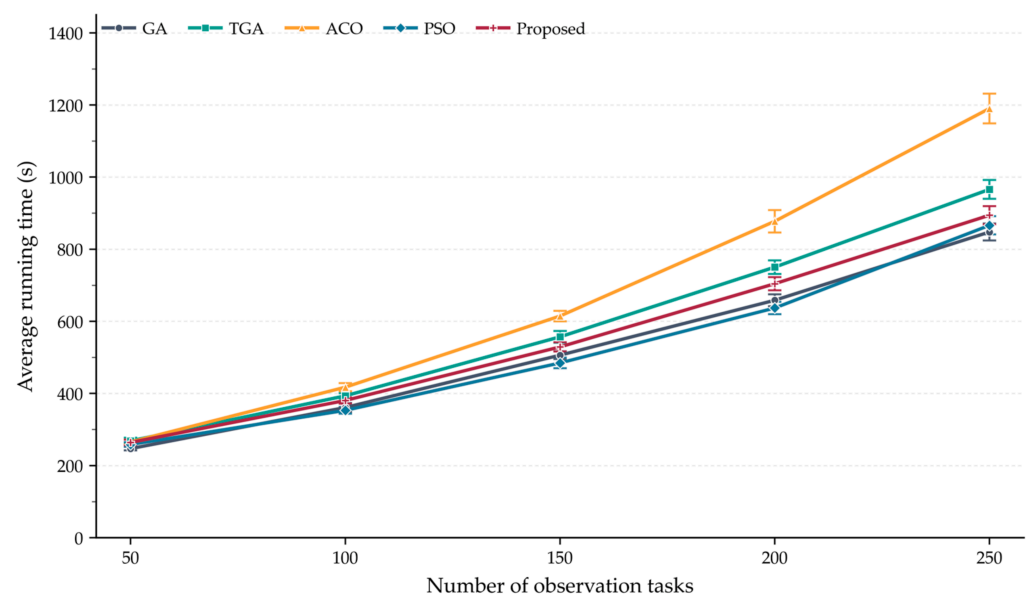


Figure 8. Average running time of different algorithms under different task scales in the LEO target scenario.

As shown in Figure 9, the running times of all algorithms in the mixed MEO–HEO scenario increase as the number of candidate tasks grows from 50 to 250, with a more pronounced increase toward the upper end of the tested range. This behavior is mainly associated with the longer visibility windows of MEO and HEO targets. As the number of tasks increases, prolonged ground-station occupancy, greater overlap among visibility windows, and more task conflicts add to the computational effort required for feasibility checking, candidate adjustment, and iterative search. With 250 candidate tasks, the proposed algorithm has an average running time of 1030.572 s, which is 8.29% and 25.46% lower than those of TGA and ACO, respectively. Its running-time standard deviation is 33.594 s, indicating reasonably stable computational cost across independent runs. Together with the preceding scheduling results, these findings show that the proposed algorithm maintains high total observation profit and task completion rate in the mixed MEO–HEO scenario. Overall, although the proposed algorithm is not the fastest in every case, its additional computational cost remains acceptable relative to its scheduling performance, indicating a favorable balance among solution quality, computational efficiency, and stability across the tested task loads.

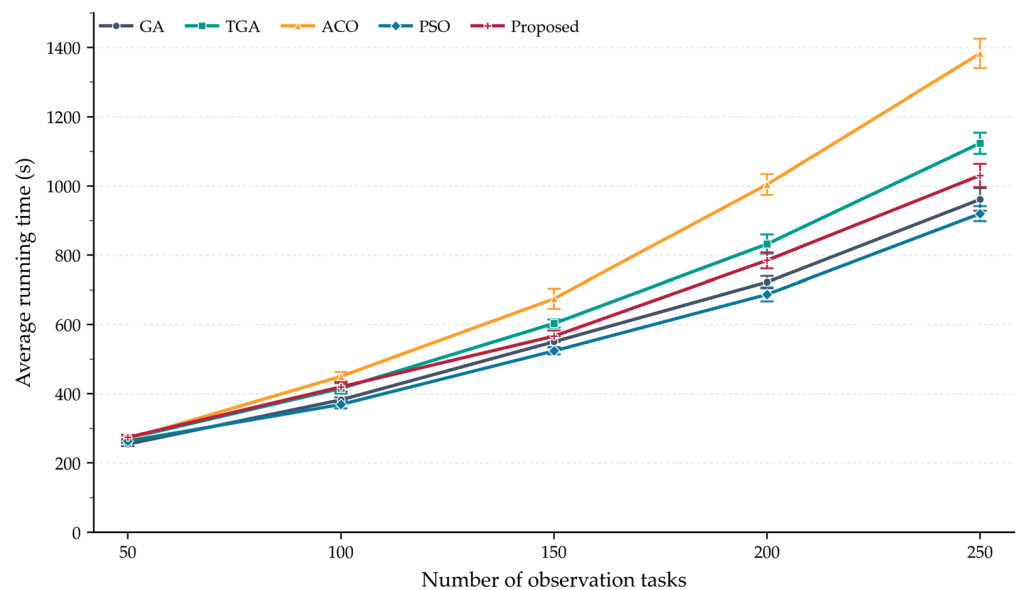


Figure 9. Average running time of different algorithms under different task scales in the MEO–HEO target scenario.

5. Conclusions

This study addresses the ground-based optical station resource scheduling problem for satellite observation tasks by establishing a corresponding resource scheduling model and proposing a dual-tabu-memory-driven station-structured genetic algorithm. To overcome the limitations of conventional genetic algorithms, such as insufficient utilization of intra-station task structures and repetitive searches, improvements are introduced in chromosome encoding, initial population generation, crossover, and mutation operations. The station-group-based encoding enhances the correspondence between chromosome structures and actual resource configurations, the heuristic initialization strategy balances initial solution quality and population diversity, the structured crossover and neighborhood-enhanced mutation improve local search capability, and the local and global tabu memory mechanisms further reduce repetitive searches. Experimental results demonstrate that the proposed method can obtain high-quality feasible scheduling solutions under different population sizes, task scales, and orbital scenarios, while achieving favorable overall performance in terms of total observation profit, task completion rate, run-to-run stability, and computational efficiency. Nevertheless, this study still has several limitations. The current research is primarily conducted under deterministic observation conditions and does not consider uncertainties such as weather variations, equipment failures, and visibility-window prediction errors; therefore, the adaptability of the proposed method to dynamic tasks and changing resource states requires further investigation. Furthermore, systematic quantitative ablation analyses of the key algorithmic components have not yet been conducted, and the sensitivity of critical algorithm parameters requires further evaluation. Future work will incorporate practical factors such as weather conditions, equipment states, and visibility-window uncertainties, and investigate adaptive scheduling and robust optimization strategies for dynamic task environments. Meanwhile, module-level ablation experiments and parameter sensitivity analyses will be conducted to further evaluate the contributions of individual components and the influence of key parameters, thereby improving the applicability and reliability of the proposed algorithm in complex real-world observation environments.

Author Contributions: Conceptualization, S.Z., T.Z. and W.C.; Investigation, S.Z., T.Z. and W.C.; Methodology, S.Z. and T.Z.; Project administration, S.Z. and T.Z.; Supervision, S.Z., T.Z. and W.C.;

Writing—review and editing, S.Z., T.Z. and W.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The datasets generated during the study are available from the corresponding author on reasonable request.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

OMRS	Optical Measurement Resource Scheduling
GA	Genetic Algorithm
TGA	Tabu Genetic Algorithm
PSO	Particle Swarm Optimization
ACO	Ant Colony Optimization
LEO	Low Earth Orbit
MEO	Medium Earth Orbit
HEO	High Earth Orbit
TT&C	Tracking, Telemetry and Command
NSGA-II	Non-dominated Sorting Genetic Algorithm II

References

1. Brown, N.; Arguello, B.; Nozick, L.; Xu, N. A Heuristic Approach to Satellite Range Scheduling with Bounds Using Lagrangian Relaxation. *IEEE Syst. J.* **2018**, *12*, 3828–3836. [\[CrossRef\]](#)
2. Luo, K.; Wang, H.; Li, Y.; Li, Q. High-Performance Technique for Satellite Range Scheduling. *Comput. Oper. Res.* **2017**, *85*, 12–21. [\[CrossRef\]](#)
3. Liu, Z.; Feng, Z.; Ren, Z. Route-Reduction-Based Dynamic Programming for Large-Scale Satellite Range Scheduling Problem. *Eng. Optim.* **2019**, *51*, 1944–1964. [\[CrossRef\]](#)
4. Tian, M.; Ma, G.; Huang, P.; Cheng, B.; Li, W. Optimizing Satellite Ground Station Facilities Scheduling for RSGS: A Novel Model and Algorithm. *Int. J. Digit. Earth* **2023**, *16*, 3949–3972. [\[CrossRef\]](#)
5. Lee, M.; Yu, S.; Kwon, K.; Lee, M.; Lee, J.; Kim, H. Mixed-Integer Linear Programming Model for Scheduling Missions and Communications of Multiple Satellites. *Aerospace* **2024**, *11*, 83. [\[CrossRef\]](#)
6. Petelin, G.; Antoniou, M.; Papa, G. Multi-Objective Approaches to Ground Station Scheduling for Optimization of Communication with Satellites. *Optim. Eng.* **2023**, *24*, 147–184. [\[CrossRef\]](#)
7. Ronen, R.; Ben-Moshe, B. Maximizing Nanosatellite Throughput via Dynamic Scheduling and Distributed Ground Stations. *Sensors* **2025**, *25*, 7538. [\[CrossRef\]](#) [\[PubMed\]](#)
8. Zhang, Z.; Zhang, N.; Feng, Z. Multi-Satellite Control Resource Scheduling Based on Ant Colony Optimization. *Expert Syst. Appl.* **2014**, *41*, 2816–2823. [\[CrossRef\]](#)
9. Zhang, Z.; Hu, F.; Zhang, N. Ant Colony Algorithm for Satellite Control Resource Scheduling Problem. *Appl. Intell.* **2018**, *48*, 3295–3305. [\[CrossRef\]](#)
10. Li, Y.; Wang, R.; Liu, Y.; Xu, M. Satellite Range Scheduling with the Priority Constraint: An Improved Genetic Algorithm Using a Station ID Encoding Method. *Chin. J. Aeronaut.* **2015**, *28*, 789–803. [\[CrossRef\]](#)
11. Song, Y.-J.; Zhang, Z.-S.; Song, B.-Y.; Chen, Y.-W. Improved Genetic Algorithm with Local Search for Satellite Range Scheduling System and Its Application in Environmental Monitoring. *Sustain. Comput. Inform. Syst.* **2019**, *21*, 19–27. [\[CrossRef\]](#)
12. Chen, M.; Wen, J.; Song, Y.-J.; Xing, L.; Chen, Y. A Population Perturbation and Elimination Strategy Based Genetic Algorithm for Multi-Satellite TT&C Scheduling Problem. *Swarm Evol. Comput.* **2021**, *65*, 100912. [\[CrossRef\]](#)
13. Zhang, J.; Xing, L. An Improved Genetic Algorithm for the Integrated Satellite Imaging and Data Transmission Scheduling Problem. *Comput. Oper. Res.* **2022**, *139*, 105626. [\[CrossRef\]](#)
14. Xu, L.; Yu, C.; Wu, B.; Gao, M. A Hybrid Genetic Algorithm for Ground Station Scheduling Problems. *Appl. Sci.* **2024**, *14*, 5045. [\[CrossRef\]](#)
15. Wu, W.; Wu, X.; Liu, F.; Liu, S.; Wang, T.; Xing, Z. Single-Pass Imaging Planning with a Hybrid Genetic Tabu Search Algorithm for Agile Earth Observation Satellites. *J. Aerosp. Eng.* **2025**, *38*, 04025044. [\[CrossRef\]](#)

16. Du, Y.; Xing, L.; Zhang, J.; Chen, Y.; He, Y. MOEA Based Memetic Algorithms for Multi-Objective Satellite Range Scheduling Problem. *Swarm Evol. Comput.* **2019**, *50*, 100576. [[CrossRef](#)]
17. Zhang, Z.; Chen, Y.; He, L.; Xing, L.; Tan, Y. Learning-Driven Many-Objective Evolutionary Algorithms for Satellite-Ground Time Synchronization Task Planning Problem. *Swarm Evol. Comput.* **2019**, *47*, 72–79. [[CrossRef](#)]
18. Zhang, J.; Xing, L.; Peng, G.; Yao, F.; Chen, C. A Large-Scale Multiobjective Satellite Data Transmission Scheduling Algorithm Based on SVM+NSGA-II. *Swarm Evol. Comput.* **2019**, *50*, 100560. [[CrossRef](#)]
19. Song, Y.-J.; Ma, X.; Li, X.-J.; Xing, L.-N.; Wang, P. Learning-Guided Nondominated Sorting Genetic Algorithm II for Multi-Objective Satellite Range Scheduling Problem. *Swarm Evol. Comput.* **2019**, *49*, 194–205. [[CrossRef](#)]
20. Xiong, M.; Xiong, W.; Liu, Z. Evolutionary Multiobjective Satellite Range Scheduling with Learning-Guided Population Generation. *IEEE Access* **2022**, *10*, 84664–84679. [[CrossRef](#)]
21. Xiong, M.; Xiong, W.; Liu, Z. A Co-Evolutionary Algorithm with Elite Archive Strategy for Generating Diverse High-Quality Satellite Range Schedules. *Complex Intell. Syst.* **2023**, *9*, 5157–5172. [[CrossRef](#)]
22. Yao, X.; Pan, X.; Zhang, T.; Li, W.; Wang, J. Knowledge-Guided Evolutionary Algorithm for Multi-Satellite Resource Scheduling Optimization. *Future Gener. Comput. Syst.* **2024**, *156*, 130–141. [[CrossRef](#)]
23. Song, Y.; Suganthan, P.N.; Pedrycz, W.; Yan, R.; Fan, D.; Zhang, Y. Energy-Efficient Satellite Range Scheduling Using a Reinforcement Learning-Based Memetic Algorithm. *IEEE Trans. Aerosp. Electron. Syst.* **2024**, *60*, 4073–4087. [[CrossRef](#)]
24. Wang, T.; Gu, Y.; Wang, H.; Wu, G. Adaptive Variable Neighborhood Search Algorithm with Metropolis Rule and Tabu List for Satellite Range Scheduling Problem. *Comput. Oper. Res.* **2024**, *170*, 106757. [[CrossRef](#)]
25. Wang, Y.; Dong, J.; Xu, X.; Wang, H.; Long, J. Solving the Integrated-Tasks Satellite Range Scheduling Problem: A Surrogate-Assisted Variable Neighborhood Search Approach. *Transp. Res. C Emerg. Technol.* **2025**, *180*, 105312. [[CrossRef](#)]
26. Liu, Z.; Liu, J.; Liu, X.; Yan, J.; Cheng, Y.; Chen, Y. An Iterated Adaptive Large Neighborhood Search Algorithm for the Large-Scale Communication Satellite Range Scheduling Problem. *Expert Syst. Appl.* **2025**, *278*, 127377. [[CrossRef](#)]
27. Hu, X.; Zhu, W.; An, B.; Jin, P.; Xia, W. A Branch and Price Algorithm for EOS Constellation Imaging and Downloading Integrated Scheduling Problem. *Comput. Oper. Res.* **2019**, *104*, 74–89. [[CrossRef](#)]
28. He, L.; Liang, B.; Li, J.; Sheng, M. Joint Observation and Transmission Scheduling in Agile Satellite Networks. *IEEE Trans. Mob. Comput.* **2022**, *21*, 4381–4396. [[CrossRef](#)]
29. He, C.; Dong, Y. Multi-Satellite Observation-Relay Transmission-Downloading Coupling Scheduling Method. *Remote Sens.* **2023**, *15*, 5639. [[CrossRef](#)]
30. Li, P.; Wang, H.; Zhang, Y.; Pan, R. Mission Planning for Distributed Multiple Agile Earth Observing Satellites by Attention-Based Deep Reinforcement Learning Method. *Adv. Space Res.* **2024**, *74*, 2388–2404. [[CrossRef](#)]
31. Li, S.; Wang, G.; Chen, J. AEM-D3QN: A Graph-Based Deep Reinforcement Learning Framework for Dynamic Earth Observation Satellite Mission Planning. *Aerospace* **2025**, *12*, 420. [[CrossRef](#)]
32. Zhu, G.; Wang, G.; Fu, W.; Han, C. Dynamic Task Planning for Heterogeneous Platforms via Spatio-Temporal and Capability Dual-Driven Framework. *Electronics* **2026**, *15*, 202. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.