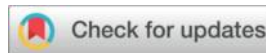




Agile Satellite Mission Planning Based on Grouped Adaptive Reward-Penalty Grey Wolf Optimizer



Lin Cao¹, Chen Wang¹, Jinxiao Wang^{2*}, Benkui Zhang²,
Pengfei Gao¹, Kangning Du¹, Yu Liu²

¹Key Laboratory of Information and Communication Systems, Ministry of Information Industry, Beijing Information Science and Technology University, Beijing, 100192, China.

²Key Laboratory of Target Cognition and Application Technology, Aerospace Information Research Institute, Chinese Academy of Sciences, Beijing, 100190, China.

Lin Cao: CharLin26@163.com
ChenWang: wangchen_0131@163.com
Jinxiao Wang: wangjx_group@163.com
Benkui Zhang: zhangbk0566@126.com
Pengfei Gao : gpf_2024@163.com
Kangning Du: kangningdu@bistu.edu.cn
Yu Liu : liuyu02@aircas.ac.cn

*Corresponding author: Jinxiao Wang. wangjx_group@163.com +Newpoint88

Acknowledgements: This work was supported by the National Natural Science Foundation of China under Grant 62201066 and Grant U20A20163. 国家自然科学基金资助项目62201066和U20A20163

Abstract

Efficient mission planning for agile Earth observation satellites is crucial for optimizing resource utilization. Traditional scheduling methods often falter under complex constraints, struggling to balance solution quality with computational efficiency. To address this challenge, this paper introduces a novel Grouped Adaptive Reward-Penalty Grey Wolf Optimizer (GARP-GWO) tailored for agile Earth observation satellite mission planning. The method comprises three key innovations: Firstly, a dynamic conflict grouping method is proposed to partition satellite observation arcs into independent groups, thereby reducing complexity by efficiently resolving intra-group scheduling conflicts. Secondly, a dual-time heuristic method that employs dynamic urgency scoring for both cut-of and start times is introduced, enabling the rapid generation of high-quality initial solutions. Lastly, an adaptive reward-punishment search strategy enhances the Grey Wolf Optimizer by incorporating leader-guided exploration, dynamic step adjustment, and exponential feedback, striking a balance between exploration and exploitation. Experimental results highlight the proposed method's superior performance, yielding higher task completion rates and faster solution times

compared to state-of-the-art techniques. This research makes a significant contribution to the field of satellite mission planning under intricate constraints, advancing the capabilities of agile Earth observation satellites.

Keywords: Agile Satellite Mission Planning, Dynamic Conflict Grouping, Heuristic Initialization, Grey Wolf Optimizer

1 INTRODUCTION

Rapid advancements in satellite technology have significantly enhanced the observational capabilities of Agile Earth Observation Satellites (AEOS) [1]. Unlike traditional Earth Observation Satellites (EOS) [2], AEOS can adjust their attitude along three axes—roll, pitch, and yaw—enabling more flexible and precise targeting. This agility allows AEOS to capture dynamic, high-resolution data over a wider range of targets, making them indispensable for modern Earth observation missions. However, the growing demand for high-frequency, high-precision Earth observation data, coupled with the increasing number of AEOS, has intensified the need for efficient mission scheduling. Effective satellite mission planning now faces a critical challenge, requiring the coordination of numerous complex factors, including satellite attitude control, resource allocation, and communication constraints [3], [4].

In the Agile Earth Observation Satellite Scheduling Problem (AEOSSP), each observation task is defined by an Observation Window (OW), which represents the feasible time interval within which a target can be observed. This OW is a subset of a larger Visible Time Window (VTW), which delineates the potential periods during which a satellite can maintain line-of-sight with a given target. Extending the VTW can increase observation opportunities, but it also significantly complicates the scheduling process. Overlapping VTWs lead to conflicts that necessitate frequent satellite attitude adjustments, increasing transition times and creating resource competition. This not only reduces the number of feasible solutions but also places significant stress on satellite systems, potentially impacting the overall mission efficiency [5].

Solutions to the AEOSSP are generally divided into two main categories: exact algorithms and heuristic algorithms.

Exact algorithms, such as Branch and Bound (B&B), cutting plane techniques, and dynamic programming, are well-suited for finding optimal solutions to small-scale problems due to their precise mathematical formulations. Bianchessi et al. [6] introduced a column generation approach to solve the linear relaxation of the AEOSSP, producing upper-bound solutions via constraint relaxation. This method theoretically supports multi-satellite, multi-orbit, and multi-user scheduling, but its computational efficiency drastically decreases as the problem size increases, rendering it impractical for large-scale instances. Valicka et al. [7] proposed a deterministic Mixed-Integer Linear Programming (MILP) model that ensures optimality and provides a detailed examination of VTW discretization. However, this approach is limited by its high computational costs. Chu et al. [8] developed a B&B algorithm specifically designed for the AEOSSP, achieving optimal solutions but only feasible for problems with up

to 55 targets due to its exponential computational complexity. To improve scalability, Chu et al. [9] also introduced an Anytime B&B algorithm, which provides intermediate solutions while continuously refining them over time. Nevertheless, even this approach struggles with large-scale problem instances.

To address the limitations of exact methods, researchers have increasingly turned to heuristic and metaheuristic algorithms, which can produce high-quality approximate solutions in a fraction of the time required by exact methods. By trading strict optimality for faster convergence and better scalability, these approaches become more practical for solving large-scale AEOSSP instances. For example, Lemaître et al. [10–12] leveraged the French Pleiades satellite mission as a testbed to develop a comprehensive understanding of the AEOSSP. They employed various techniques, including greedy algorithms, dynamic programming, constraint programming, and local search methods, to tackle the optimization challenges posed by agile Earth observation satellites. Subsequently, Wolfe and Sorensen [13] expanded on this work by incorporating a genetic algorithm that prioritizes high-value tasks, optimizing task sequencing with a priority scheduling method. This domain-specific approach significantly improved scheduling efficiency, offering valuable insights for satellite mission management under complex constraints.

Recently, researchers have advanced the heuristic solutions for AEOSSP by focusing on multi-objective modeling, hierarchical optimization, dynamic neighborhood exploration, and hybrid strategies. For instance, Wang et al. [14] redefined the AEOSSP as an overbooking problem, utilizing a directed graph representation where each candidate observation target is represented as a node. This graph-based method allows for more precise conflict resolution and task prioritization, effectively capturing the intricate interdependencies among multiple satellites and enhancing overall task allocation efficiency. In a follow-up study, Wang et al. [15] extended this framework to multi-satellite scheduling by proposing a distributed task allocation strategy grounded in complex network theory. This approach decomposes the global problem into smaller, more manageable subproblems, greatly improving scalability.

Building on these advancements, Liu et al. [16] introduced an Adaptive Large Neighborhood Search (ALNS) algorithm tailored for single-satellite scheduling. This method dynamically adjusts the balance between destruction and repair operators, enabling the recalibration of attitude transitions and alignment of feasible observation windows in each iteration. The inclusion of rapid feasibility checks accelerates solution verification, making the ALNS algorithm highly effective for medium-sized AEOSSP instances. Peng et al. [17] further refined task sequencing with an enhanced insertion procedure that considers time-dependent margins and attitude transition times. Meanwhile, He et al. [18] pursued a bilateral optimization and divide-and-conquer strategy, dividing the overall problem into a high-level task assignment layer and a lower-level single-satellite scheduling layer. This hierarchical approach is implemented through an Adaptive ALNS (A-ALNS) algorithm, incorporating five distinct allocation operators that significantly improve task distribution efficiency. Recently, Chang et al. [19] developed a hybrid approach combining ALNS with NSGA-II. Although their method was not extended to large-scale scheduling scenarios, they successfully introduced novel operators that significantly improved algorithm efficiency.

Most notably, Peng et al. [20] introduced the Greedy Randomized Iterated Local Search (GRILS) algorithm, which combines instantaneous transition time modeling with specialized insertion operators. This method integrates rapid feasibility checks and advanced task allocation procedures, thereby significantly improving both the quality and computational efficiency of solutions compared to earlier ALNS-based approaches. Consequently, GRILS is now considered one of the premier heuristic methods for AEOSSP, especially for large-scale problems.

Despite such advancements, current heuristic algorithms still encounter significant challenges when applied to larger AEOSSP instances. The primary issues include the lack of guidance in the search process and inefficient management of large solution spaces. Most existing algorithms, including GRILS, rely on iterative destruction-repair mechanisms where task removal is often arbitrary and not necessarily tied to task priority or conflict intensity. This can disrupt high-quality solutions and necessitate numerous additional iterations for recovery. Furthermore, traditional methods struggle to capture complex task dependencies or effectively decompose vast solution spaces, leading to exponential growth in candidate VTWs as the problem size increases. This, in turn, significantly escalates computational costs.

To tackle these challenges, this paper introduces the Group-based Adaptive Reward-Penalty Grey Wolf Optimizer (GARP-GWO). Drawing inspiration from the social hierarchy and collaborative hunting strategies of grey wolves [21], the GARP-GWO algorithm incorporates three key innovations aimed at overcoming the limitations of traditional heuristic methods. These innovations include:

1. The GWO algorithm is applied to the AEOSSP for the first time. This marks a significant step in exploring new optimization approaches for this domain, offering a fresh perspective on solving complex satellite scheduling problems.
2. This study introduces the Dynamic Conflict Grouping Method (DCGM) to AEOSSP. By partitioning continuous observation arcs into independent "wolf packs" through the resolution of time conflicts and attitude transition constraints, DCGM greatly enhances computational efficiency, particularly for large-scale problems.
3. We propose a novel strategy for generating high-quality initial solutions for AEOSSP. This approach integrates deadline urgency assessment, dynamic start-time scoring, and resource allocation sorting, effectively addressing the complexities of scheduling and resource management tasks.
4. An Adaptive Reward-Penalty Search (ARPS) strategy is proposed, which employs head-wolf-guided local search and exponential feedback to dynamically adjust search intensity. This strategy effectively avoids premature convergence and improves the solution quality, especially for large-scale tasks.

2 Multi-Satellite AEOSSP Model

2.1 Model Assumptions

This paper makes the following assumptions for the AEOSSP problem.

1. There are sufficient ground station and relay satellite resources.

Table 1 Parameter Definitions

T	Set of targets
T_i	Target T_i
u_i	Number of VTWs of target T_i
p_i	Profile of target T_i
d_i	Duration of target T_i
S	Set of satellites
M	Set of Wolf packs
M_j	Wolf M_j
x_j^i	If target T_i is completed by Wolf M_j , $x_j^i = 1$
$[st_j, et_j]$	Start and end time of Wolf M_j
n_j	Number of VTWs in Wolf M_j
c_j	The average resource allocation rate of Wolf M_j
VTW_i^j	VTW of Wolf M_j for target T_i
ET_i^j	Exclusive time of VTW_i^j
$COST_i^j$	Cost of VTW_i^j
a_i	Indicates whether target T_i belongs exclusively to M_j
vs_i^j, ve_i^j, vts_i^j	Start, top, and end time of VTW_i^j
s_i^j	Score of VTW_i^j
OW_i^j	Selected OW of VTW_i^j
os_i^j	Start time of OW_i^j
$\phi_i^j, \theta_i^j, \psi_i^j$	Roll, pitch, yaw angles of OW_i^j
$\Delta g(OW_m^j, OW_n^j)$	Difference in observation angle between OW_m and OW_n
$tran(\Delta g)$	Corresponds to the required transition time
$Tran_{max}$	The maximum required transition time

2. Each AEOS can observe only one target at a time.
3. Satellites have sufficient capabilities and storage capacity.
4. Each target can be observed at most once.
5. All OWs must be subsets of their corresponding VTWs.
6. A sufficient attitude transition time must be allocated between consecutively scheduled observation windows.

The optimization objective of this paper is to maximize the number of observation tasks. Table 1 provides the symbols and definitions used in this paper.

2.2 AEOSSP Model

The mathematical formulation of the AEOSSP model in this paper is as follows:

$$\text{Maximize } \sum_{i \in T} \sum_{j \in M} p_i \cdot x_i^j \quad (1)$$

$$\text{subject to: } \sum_{j \in M} x_i^j \leq 1, \forall i \in T. \quad (2)$$

$$x_i^j \in \{0, 1\}, \forall i \in T \quad (3)$$

$$os_j^i \in \mathbb{R}, \forall i \in T, j \in M \quad (4)$$

$$x_i^j \cdot (os_j^i - os_j^i) \geq 0, \forall i \in T, j \in M \quad (5)$$

$$x_i^j \cdot (ve_j^i - d_i - os_j^i) \geq 0, \forall i \in T, j \in M \quad (6)$$

$$os_m^i + d_i + \text{tran}(\Delta g(OW_m^i, OW_n^i)) \geq os_n^i, \quad (7)$$

$$\forall m, n \in T, j \in M.$$

The optimization objective is defined in (1), where the goal is to maximize the total value of scheduled targets based on their assigned observation profiles. To avoid assigning the same target multiple times, the multi-orbit uniqueness constraint (2) ensures that each target is observed at most once across all arcs. The binary decision variable x_i^j , defined in (3), indicates whether target T_i is assigned to arc M_j . Specifically, $x_i^j = 1$ if the assignment exists; otherwise, it is set to 0.

Once a target is scheduled, (4) requires that the actual observation start time os_j^i must not be earlier than the earliest available time vs_j^i in the corresponding VTW_j^i . This constraint ensures that the scheduled observation respects the orbital feasibility by aligning the observation with the satellite's available viewing periods. Similarly, (5) ensures that the observation finishes within the valid time window by requiring the end time $os_j^i + d_i$ not to exceed the VTW's upper limit ve_j^i .

To avoid scheduling conflicts between consecutive tasks, (6) imposes a non-overlapping condition between consecutive tasks on the same arc, guaranteeing sufficient spacing between observations and reducing the risk of conflicts.

Finally, the transition time between observations is handled by (7), which models the required adjustment time as a function of the angular distance between two consecutive observation windows. The angle difference is represented as $\Delta g(OW_m^i, OW_n^i)$, and the corresponding transition time is defined as $\text{tran}(\Delta g)$. This approach captures the physical limitations of satellite maneuvers, ensuring that attitude transitions remain feasible within the system's operational constraints. According to the method outlined in [16], the transition time is computed as:

$$\text{tran}(\Delta g) = \begin{cases} 11.66, & \Delta g \leq 10^\circ \\ a_1 + \frac{\Delta g}{v_1}, & 10^\circ < \Delta g \leq 30^\circ \\ a_2 + \frac{\Delta g}{v_2}, & 30^\circ < \Delta g \leq 60^\circ \\ a_3 + \frac{\Delta g}{v_3}, & 60^\circ < \Delta g \leq 90^\circ \\ a_4 + \frac{\Delta g}{v_4}, & \Delta g > 90^\circ. \end{cases} \quad (8)$$

where, v_1, v_2, v_3, v_4 represent four different angular velocities, corresponding to different values of changes in viewing angle. a_1, a_2, a_3, a_4 are the constant terms associated with the angular velocities. Δg represents the change in viewing angle during the transition, and its calculation equation is (9).

$$\Delta g(OW_m^i, OW_n^i) = |\phi_m^i - \phi_n^i| + |\theta_m^i - \theta_n^i| + |\psi_m^i - \psi_n^i| \quad (9)$$

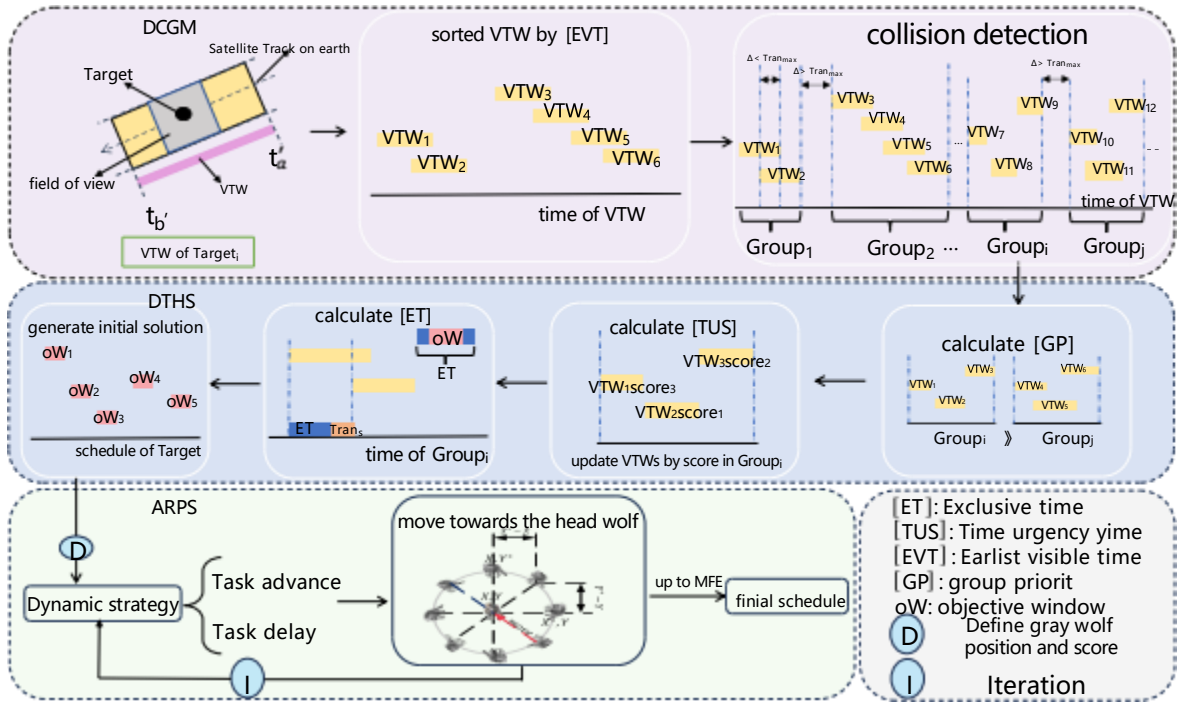


Fig. 1 Flowchart of GARP-GWO algorithm.

If $tran(\Delta g)$ does not satisfy (7), it indicates that the two OWs can't be scheduled simultaneously, and one of them needs to be deleted or reprocessed.

3 Grouping Adaptive Reward-Penalty Grey Wolf Optimizer

The proposed GARP-GWO framework, illustrated in Fig.1, consists of a DCGM for large-scale tasks, a DTHS generation strategy based on Earliest Deadline First (EDF), and an ARPS strategy.

3.1 Dynamic Conflict Grouping Method

In AEOSSP, the optimization objective is to maximize the task completion rate by reasonably allocating VTWs, adhering to the following constraints:

1. VTWs of a single satellite need to satisfy the same satellite must meet the requirements criteria of non-overlapping time intervals and attitude transition time.
2. Each task target can be executed by at most one satellite.

Given the aforementioned constraints, conventional Grey Wolf Optimizer (GWO) approaches generally model the entire solution set as a single wolf pack. However,



Fig. 2 Schema of wolf packs in the GARP-GWO.

when directly applied to the Attitude-constrained Earth Observation Satellite Scheduling Problem (AEOSSP), this approach encounters significant challenges in detecting global conflicts and falls short in effectively meeting the stringent attitude transition constraints between observation windows. To address these issues, this paper introduces a Dynamic Conflict Grouping Method (DCGM), which divides a satellite's continuous observation arcs into multiple wolf packs based on conflict relationships, as illustrated in Fig. 2.

In this context, $Tran_{max}$ denotes the maximum allowable transition time required for satellite attitude changes. If the time interval Δ between two consecutive Viewing Time Windows (VTWs) satisfies the condition $\Delta \leq Tran_{max}$, they are considered to belong to the same wolf pack. Within each wolf pack, both temporal and attitude transition conflicts are resolved locally, enabling more efficient conflict management. Significantly, there are no temporal conflicts between different wolf packs, thereby effectively decomposing the global optimization task into several smaller, independent subproblems. This strategy notably reduces computational complexity and enhances the scalability of the GWO algorithm for large-scale AEOSSP instances.

The DCGM method leverages the idea of grouping related observation tasks into smaller, manageable units (wolf packs), where conflicts are primarily local. This approach enables each wolf pack to focus on its own scheduling issues while minimizing interference from other packs. As a result, the optimization process becomes more modular and less prone to the computational bottlenecks that often occur with monolithic, centralized scheduling algorithms. This modularization not only improves the efficiency of conflict resolution but also allows for a more flexible and adaptive implementation, making it easier to scale the solution to handle larger and more complex satellite scheduling scenarios.

Moreover, by dynamically adjusting the grouping of observation windows based on changing conditions and constraints, the DCGM method offers a more robust and responsive solution framework. This adaptability is especially important in real-world environments where observation tasks and constraints can evolve over time. The dynamic grouping mechanism ensures that the solution remains optimized and feasible even as new data and requirements are introduced into the system. Consequently, the DCGM-enhanced GWO approach is better suited to handle the complexities and challenges inherent in AEOSSP, leading to more effective and efficient satellite observation scheduling.

In the GARP-GWO algorithm, a wolf pack consists of mutually conflicting VTWs. However, in practical mission planning, a satellite typically does not need to occupy the entire VTW to capture a target. Thus, even when two VTWs overlap temporally, observing their associated targets remains feasible with reasonable scheduling. Thus,

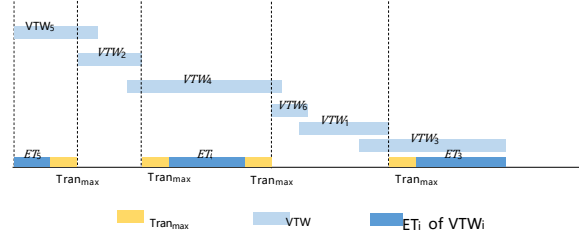


Fig. 3 Schema of exclusive time

each VTW in a wolf pack can be assigned a unique time segment, termed the exclusive time, as shown in Fig.3, which avoids any overlap with others.

3.2 Dual-Time Heuristic Initial Solution generation strategy

The AEOSSP is characterized by strong constraints, multi-objective optimization, and high-dimensional discreteness. As the task scale increases, the solution space expands exponentially, making efficient search increasingly challenging. In this context, the quality of the initial solution plays a critical role in determining both the convergence speed and the final solution quality of heuristic algorithms. However, conventional random initialization methods often fail to respect essential constraints, such as time window overlaps and attitude transitions, leading to suboptimal solutions and increased search costs.

To address this problem, this paper proposes a Dual-Time Heuristic Initial Solution (DTHS) generation strategy that incorporates both deadline urgency and start time prioritization. This approach is further enhanced by a wolf pack resource allocation rate sorting mechanism, designed to produce high-quality initial solutions that better align with the problem's physical constraints. In this strategy, a deadline urgency score is introduced to quantify the time pressure of each VTW, reflecting the relative importance of scheduling each observation task within its available window. The score is defined as:

$$s_i^j = L \times \frac{ve_i^j - st_j}{et_j - st_j} - 0.5L. \quad (10)$$

Here $L = n_j \times D$ is the normalization coefficient, n_j represents the number of VTWs in the wolf pack M_j , D is a tuning constant, ve_i^j is the latest effective observation time for task T_i . st_j and et_j denote the start time and end time of the wolf pack M_j , respectively. When $ve_i^j \rightarrow et_j$, the score s_i^j approaches the maximum $0.5L$, denoting high urgency and priority in scheduling this VTW. Conversely, when $ve_i^j \rightarrow st_j$, the score approaches the minimum $-0.5L$, suggesting that this VTW can be scheduled later.

To further mitigate the risk of task accumulation, a start time urgency correction term is introduced to encourage the early scheduling of observation tasks. The revised score reflects the combined urgency of both the deadline and the start time:

$$s_i^j = s_i^j + 0.5L. \quad (11)$$

To optimize resource allocation among wolf packs, the average resource allocation rate is defined to quantify the resource availability of each wolf pack, prioritizing those with abundant resources. The equation is given by:

$$c_j = \frac{et_j - st_j}{\sum d_i a_i u_i^j}. \quad (12)$$

If M_j can complete target T_i , then $u_i^j = 1$, otherwise $u_i^j = 0$. If target T_i is exclusive to M_j , then $a_i = 0$, otherwise $a_i = 1$.

The pseudocode for the DTHS strategy is presented in Algorithm 1.

Algorithm 1 DTHS generation strategy pseudocode

Require: Task set T , wolf pack set M
Ensure: Initial scheduling scheme S_{init}

- 1: Initialize $S_{init} \leftarrow \emptyset$
- 2: **for** each wolf pack $M_j \in M$ **do**
- 3: **for** each VTW $VTW_i^j \in M_j$ **do**
- 4: Calculate deadline urgency score s_i^j using (10)
- 5: Add start time urgency correction using (11)
- 6: **end for**
- 7: Sort VTWs in M_j in descending order of s_i^j
- 8: **end for**
- 9: Calculate average resource allocation rate c_j for each wolf pack M_j using (12)
- 10: Sort wolf packs M in descending order of c_j
- 11: **for** each wolf pack $M_j \in M$ (in sorted order) **do**
- 12: **for** each VTW $VTW_i^j \in M_j$ (in sorted order) **do**
- 13: **if** VTW_i^j does not conflict with scheduled VTWs **then**
- 14: Assign VTW_i^j to S_{init}
- 15: **end if**
- 16: **end for**
- 17: **end for**
- 18: **return** S_{init}

3.3 Adaptive Reward-Penalty Search Strategy

The Grey Wolf Optimizer (GWO) is inspired by the social hierarchy and hunting strategies of grey wolf packs. In the context of optimization, the α , β , and δ wolves act as leaders, guiding the rest of the pack toward the optimal solution, akin to tracking prey. However, in satellite mission planning, the discrete and high-dimensional nature of the task scheduling solution space introduces significant challenges. This leads to differences in the scheduling solutions represented by the three leader wolves, resulting in issues such as population dispersion, slow convergence, and a tendency to get trapped in local optima. Furthermore, the fixed-weight mechanism commonly used in traditional GWO algorithms struggles to adapt to dynamically changing resource

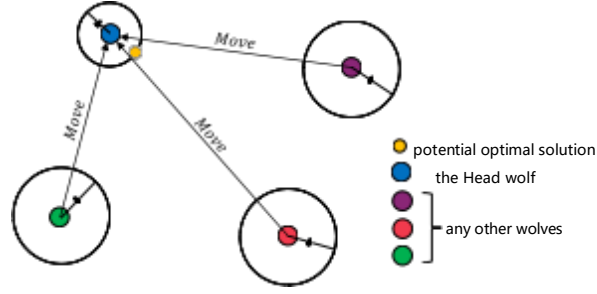


Fig. 4 Schematic diagram of the local search strategy in GARP-GWO optimization.

constraints and task priorities, thus degrading solution quality in complex real-world scenarios.

To overcome these challenges, this paper introduces an Adaptive Reward-Penalty Search (ARPS) strategy. This approach enhances the global search capability and local exploitation efficiency of the GWO by dynamically adjusting search directions based on the performance of the leading wolves. It incorporates a reward-penalty mechanism and an adaptive step size factor, allowing the algorithm to balance exploration and exploitation more effectively.

In the ARPS strategy, the best individual in the current population is designated as the “head wolf,” while the remaining wolves adjust their positions relative to this leader. As a result, the head wolf conducts a localized search within its immediate neighborhood, seeking potentially superior solutions. This process is illustrated in Fig. 4.

$$\Delta_i^j = |s_i^j(t) - s_h^j(t)|. \quad (13)$$

Based on this quantified discrepancy, the position update rule is provided in (14), where an adaptive step size factor k_t is introduced to regulate movement. The detailed formulation of k_t is elaborated in 3.4. This adaptive design enables the algorithm to adopt a larger step size in early iterations for extensive global exploration, and subsequently reduce the step size progressively in later iterations to enhance local exploitation, thereby effectively balancing exploration and exploitation.

$$s_i^j(t+1) = s_i^j(t) - \Delta_i^j \cdot k_t. \quad (14)$$

To further enhance the local search capability, this paper proposes a dynamic adjustment approach. Individuals identifying solutions superior to the current head wolf promptly update their positions and receive rewards, thereby promoting effective exploration. Conversely, individuals failing to find improved solutions incur penalties, redirecting their search trajectories away from ineffective regions. The dynamic adjustment of task priority scores mathematically implements this strategy. Tasks completed ahead of schedule receive additional reward scores defined by (15).

$$s_i^j(t+1) = s_i^j(t) - k_t \cdot D \cdot (\Delta_i^j + 1)^3. \quad (15)$$

Tasks experiencing delays or conflicts are assigned exponentially scaled penalty scores, with the calculation given in (16).

$$s_i^j(t+1) = s_i^j(t) + \frac{k_t \cdot D}{u_i}. \quad (16)$$

3.4 Exploration and Exploitation

To maintain search efficiency throughout the optimization process, the GARP-GWO algorithm dynamically adjusts the search amplitude through the adaptive factor k_t , enabling a smooth transition from global exploration to local refinement. This mechanism complements the dynamic adjustment approach by encouraging broad search behavior in early stages and progressively shifting the focus toward solution exploitation as the algorithm converges. The adjustment of k_t is governed by the following expression:

$$k_t = A \cdot B_t. \quad (17)$$

where k_t represents the value of k at the t -th iteration. Parameter A is affected by both the Non-Further Maximum Enhancements (NFME) and the iteration where the optimal solution was identified. If a new best solution is discovered at a given iteration, A is set to 0.5 to promote exploitation during that period. Otherwise, A remains 1. The parameter B_t decreases as the number of iterations increases, shifting the algorithm's focus from exploration to exploitation. It is adjusted according to the following (18):

$$B_t = 2 - \frac{t}{MFE}. \quad (18)$$

where MFE represents the Maximum Function Evaluations.

4 Experimental Results

This section evaluates the performance of the proposed algorithm through a series of comparative experiments. First, the GARP-GWO algorithm is benchmarked against the classical heuristic A-ALNS and the recent state-of-the-art heuristic GRILS on multi-satellite instances, with target sizes ranging from 100 to 2000. This comparison aims to demonstrate the feasibility and superiority of GARP-GWO in handling large-scale AEOSSP. For each task scale, ten distinct task subsets are generated, and the average performance across these instances is reported to ensure robustness and statistical reliability. Next, ablation experiments are conducted to assess the impact of different initial solution generation strategies on the overall performance of the GARP-GWO algorithm. This analysis provides insights into the effectiveness of the DTSS strategy in improving solution quality and convergence speed. Additionally, the effectiveness of the DCGM is evaluated, emphasizing its role in reducing solution space dimensionality and improving computational efficiency by isolating local conflicts and minimizing global dependency. Finally, the ARPS strategy is validated, focusing on its ability to accelerate convergence and enhance solution quality, particularly in large-scale problem instances. This comprehensive evaluation framework ensures that the proposed GARP-GWO algorithm is thoroughly tested across a range of realistic and challenging AEOSSP scenarios.

Table 2 Keplerian Orbital Elements of Satellite Constellation

Satellite	a (km)	e	i (.)	Ω (.)	(ϑ) (.)	D (.)
SAT-1	7200	0.000627	96.576	175.72	0.075	0
SAT-2	7200	0.000627	96.576	145.72	30.075	0
SAT-3	7200	0.000627	96.576	115.72	60.075	0
SAT-4	7200	0.000627	96.576	85.72	90.075	0
SAT-5	7200	0.000627	96.576	55.72	120.075	0
SAT-6	7200	0.000627	96.576	25.72	150.075	0

4.1 Test Instances

The dataset used in this study consists of 6 satellites, with their orbital parameters listed in Table 2. Each satellite has a rectangular field of view, with both the horizontal and vertical half-angles set to 7° . The maximum pitch and roll angles are fixed at 45° . These settings are consistent with those adopted in [16], [20], ensuring comparability and reproducibility with prior studies.

where a is the semi-major axis, e is the orbital eccentricity, i represents the inclination, Ω is the right ascension of the ascending node, (ϑ) denotes the argument of periapsis, and D signifies the mean anomaly.

The dataset used in this study consists of randomly generated point targets within the region of China (3°N - 53°N and 74°E - 133°E). The final number of observation targets ranges from 100 to 2000, with an interval of 100 targets. In both the baseline algorithms and the proposed approach, nearly all targets can be scheduled in global-scale instances, which does not highlight significant differences. Therefore, no comparisons are presented for global instances. The observation duration for all targets is set to 15 seconds, and the observation reward for each target is set to 1.

4.2 Comparison with State-of-the-Art Algorithms

We compare our proposed method with the state-of-the-art heuristic algorithms A-ALNS and GRILS using the same dataset. The settings for A-ALNS and GRILS follow those described in [16], [20]. In the proposed GARP-GWO method, MFE is set to 100 and $NFME$ to 5. Since the targets are randomly generated and vary across different problem instances, we generate 100 instances for each target size. The reported results represent the average performance on these instances. Since all targets have identical observation rewards, the final scheduling success rate is used to represent the total profit, denoted as F_s .

From the experimental results in Table 3 and Fig. 5 and Fig. 6, it can be observed that for small-scale tasks (100 to 500 tasks), all three algorithms—GARP-GWO, GRILS, and A-ALNS—achieve a 100% task completion rate. However, as the task scale increases, the completion rates of all three algorithms decline. Among them, A-ALNS exhibits the steepest decline, followed by GRILS, while GARP-GWO maintains the slowest rate of decline. For large-scale instances with 2000 tasks, GRILS drops to a completion rate of 47.01%, A-ALNS to 36.74%, while GARP-GWO maintains a higher completion rate of 52.17%, significantly outperforming the other two algorithms.

Table 3 Performance Comparison of Task Solving Algorithms

task scale	GRILS		A-ALNS		GARP-GWO	
	$t(s)$	$F_s(\%)$	$t(s)$	$F_s(\%)$	$t(s)$	$F_s(\%)$
100	0.04	100	0.94	100	0.38	100
200	1.08	100	38.64	98.9	1.93	100
300	3.18	100	100.17	94.13	3.46	100
400	6.24	99.97	191.55	87.24	6.15	100
500	16.43	99.42	298.17	79.11	10.04	100
600	51.09	95.8	403.70	74.09	35.55	100
700	73.572	92.17	593.19	68.58	62.4	97.41
800	99.2	84.24	794.66	63.33	77.3	92.39
900	129.32	79.23	1044.84	60.48	88.6	87.2
1000	159.7	74.08	1312.91	56.69	95.25	82.33
1100	190.57	71.25	1604.88	53.50	106.56	78.02
1200	224.24	68.21	1897.11	51.22	118.75	73.76
1300	257.06	64.17	2105.45	48.99	134.09	70.54
1400	297.6	61.24	2395.31	46.58	150.18	67.24
1500	344.67	58.31	2668.85	44.65	162.76	64.3
1600	398.4	56.07	2936.66	42.72	176.23	61.49
1700	443.13	53.41	3271.17	41.23	194.63	58.86
1800	491.01	51.16	3937.91	39.58	203.1	56.52
1900	545.11	49.12	4222.78	37.85	213.47	54.28
2000	600.45	47.01	4762.52	36.74	222.69	52.17

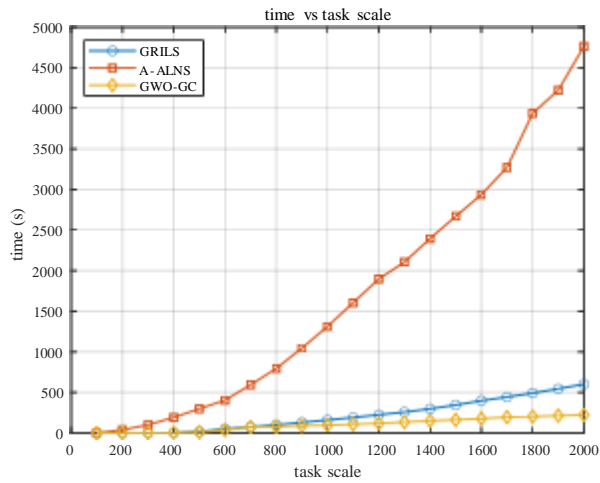


Fig. 5 Computational time performance comparison.

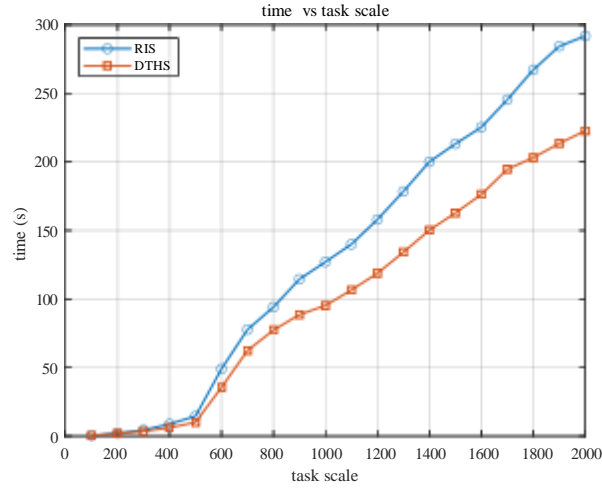


Fig. 6 Mission Completion Rate Performance Comparison.

Regarding computational time, A-ALNS consistently has the highest runtime across all task scales, reaching 4762.52 seconds for 2000 tasks, indicating poor scalability. In contrast, GRILS demonstrates the shortest computational time for small-scale tasks (100 to 300 tasks) but experiences a sharp increase as the task size grows, requiring 600.45 seconds for 2000 tasks. In comparison, GARP-GWO exhibits a more gradual and stable increase in computation time as the task size scales, requiring only 222.69 seconds for 2000 tasks. This represents a significant reduction in computational time compared to both A-ALNS and GRILS, demonstrating superior computational efficiency and scalability.

In summary, the GARP-GWO algorithm consistently achieves a better balance between task completion rate and computational efficiency. In small-scale scenarios (100 to 500 tasks), it regularly outperforms A-ALNS in both solution quality and execution time, while offering results comparable to or better than those of GRILS. As the problem size increases, the advantages of GARP-GWO become more pronounced. In the 2000-task benchmark, it improves the task completion rate by 5.16% and 15.43% compared to GRILS and A-ALNS, respectively, while reducing computational time by 62.9% and 95.3%. These results underscore the scalability and efficiency of the GARP-GWO algorithm, highlighting its potential for large-scale mission planning.

4.3 Effectiveness Verification of the Dual-Time Heuristic Initial Solution

As most existing heuristic algorithms prioritize improvements in search strategies or task models while paying limited attention to initial solution quality, the RIS is commonly adopted. To examine the effectiveness of the proposed DTHS initialization strategy, this section compares it against randomly generated initial solutions. All experimental parameters remain consistent with those in 4.1, 4.2.

Table 4 Comparison of Task Scale and Solution Performance Under Different Strategies

task scale	RIS		DTHS	
	t (s)	F_s (%)	t (s)	F_s (%)
100	0.49	100	0.38	100
200	2.12	100	1.93	100
300	4.41	100	3.46	100
400	8.82	100	6.15	100
500	14.57	100	10.04	100
600	49.04	99.68	35.55	100
700	77.70	96.17	62.40	97.41
800	94.08	91.19	77.30	92.39
900	114.48	83.20	88.60	87.20
1000	127.07	77.87	95.25	82.33
1100	139.95	73.54	106.56	78.02
1200	157.96	68.29	118.75	73.36
1300	178.43	65.05	134.09	70.54
1400	200.05	61.69	150.18	67.24
1500	213.13	57.87	162.76	64.30
1600	225.28	55.46	176.23	61.49
1700	245.38	52.18	194.63	58.86
1800	267.17	49.28	203.10	56.52
1900	284.36	47.72	213.47	54.28
2000	291.84	43.34	222.69	52.17

The completion rates F_s for different task scales are presented in Table 4, with their variation curves illustrated in Fig. 7. For task scales up to 500, both the DTHS and random initial solution (RIS) methods achieve a 100% task completion rate. However, as the task scale increases, the completion rates for both methods gradually decline, with the DTHS consistently outperforming the RIS. At the 1000-task scale, the RIS achieves a completion rate of 77.87%, while the DTHS improves this to 82.33%. For the 2000-task scale, the DTHS reaches a 52.17% completion rate, which is 8.83% higher than the 43.34% achieved by the RIS. This clearly demonstrates the significant advantage of the DTHS in enhancing the success rate of large-scale task scheduling.

Table 4 also presents the computational times t (s) for different task scales, while Fig. 8 highlights the computational growth trends for both methods. Although the computational time for both DTHS and RIS increases non-linearly as task scale expands, the DTHS consistently exhibits a slower growth rate, indicating superior scalability. For task scales below 500, the difference in computational time between the two strategies is relatively small. However, as the task scale continues to grow, the gap becomes increasingly pronounced. At the 2000-task scale, the RIS requires 291.84 seconds, whereas the DTHS significantly reduces this to 222.69 seconds, representing a 23.7% reduction. This difference further emphasizes the scalability advantage of the DTHS, as it not only reduces computational time but also maintains higher task completion rates.

Overall, these experimental results demonstrate that the proposed DTHS generation strategy offers substantial advantages in both computational efficiency and

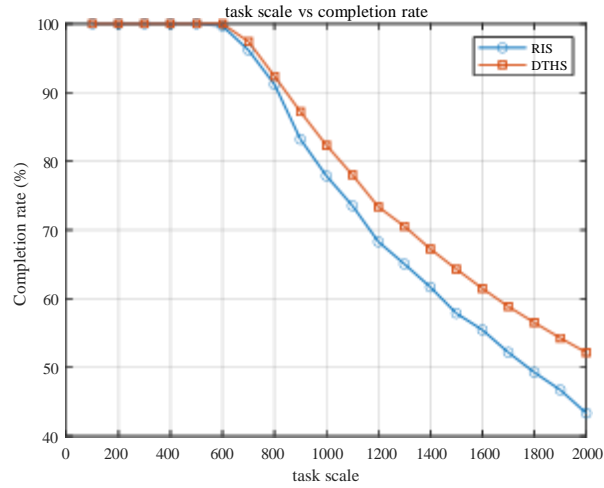


Fig. 7 Task Completion Rate: DTHS vs. RIS.

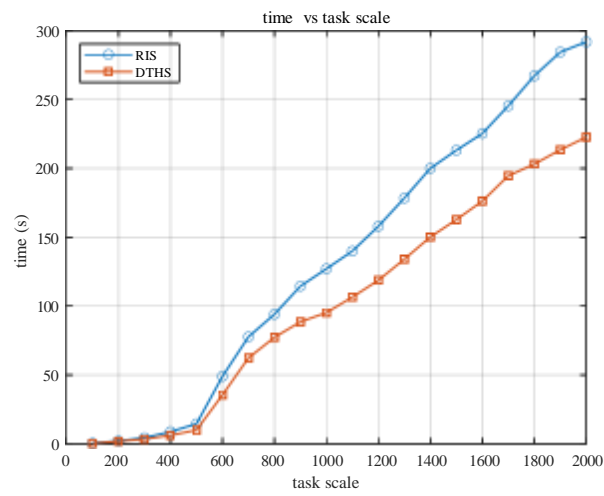


Fig. 8 Solution Time: DTHS vs. RIS.

task completion rates. In particular, for large-scale task scenarios, the DTHS effectively reduces computational resource consumption while simultaneously improving scheduling quality, making it a superior choice for large-scale AEOSSP.

4.4 Effectiveness Verification of the Dynamic Conflict Grouping Method

To verify the effectiveness of the proposed DCGM, this section compares it with an approach that does not employ a DCGM. All experimental parameters remain consistent with those in 4.1, 4.2.

Table 5 Comparison of Solution Performance Under Different Strategies

task scale	Non-DCGM		DCGM	
	t (s)	F_s (%)	t (s)	F_s (%)
100	0.66	100	0.38	100
200	2.43	100	1.93	100
300	4.86	100	3.46	100
400	9.66	100	6.15	100
500	14.57	100	10.04	100
600	51.27	96.60	35.55	100
700	79.23	93.13	62.40	97.41
800	97.61	87.40	77.30	92.39
900	106.66	82.37	88.60	87.20
1000	123.15	77.48	95.25	82.33
1100	138.49	73.02	106.56	78.02
1200	154.51	68.27	118.75	73.36
1300	175.69	65.80	134.09	70.54
1400	191.64	62.50	150.18	67.24
1500	204.52	59.71	162.76	64.30
1600	220.11	56.88	176.23	61.49
1700	240.56	54.29	194.63	58.86
1800	257.93	51.88	203.10	56.52
1900	280.43	49.85	213.47	54.28
2000	288.14	47.84	222.69	52.17

Computational time data, presented in the $t(s)$ of Table 5, reflects the performance impact of the grouping strategy under varying task scales, while Fig. 9 further confirms its superior computational efficiency in large-scale scheduling scenarios. For task scales below 500-task, the difference in computational time between the two initial solution generation strategies is not significant. However, as the task scale increases, the grouping strategy exhibits a significant advantage. When the task scale reaches 2000-task, the computational time for the non-grouping strategy is 288.14 seconds, while adopting the DCGM reduces the time to 222.69 seconds, achieving an approximately 22.7% reduction in computation time. Further observation from Fig. 9 shows that the growth curve of the DCGM is noticeably smoother, indicating that this strategy provides better computational efficiency when handling larger-scale tasks.

Task completion rates, as reported in the F_s of Table 5, provide a quantitative comparison of the two strategies, while Fig. 10 visualizes how these rates evolve with increasing task scale. For task scales below 500, both strategies maintain a 100% task completion rate. As the task scale gradually increases, the completion rates of both strategies begin to decline; however, the DCGM is still able to achieve 100% completion for tasks of scale 600, with a significantly slower decline rate. When the task scale expands to 2000, the completion rate of the non-DCGM drops to 47.84%, whereas the DCGM achieves a completion rate of 52.17%, demonstrating its clear advantage in improving task completion rates.

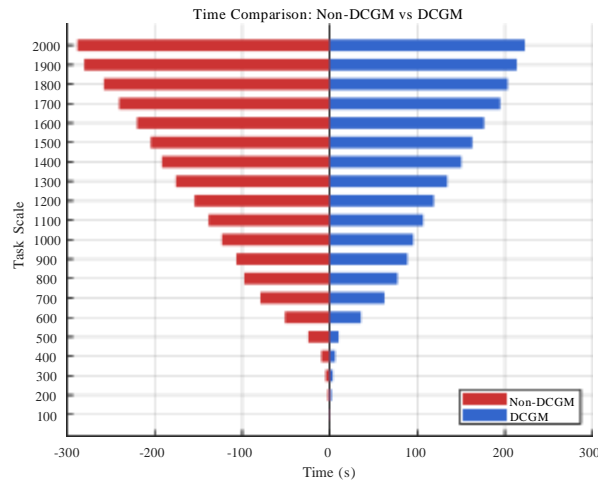


Fig. 9 Solution time comparison: DCGM vs. non-DCGM.

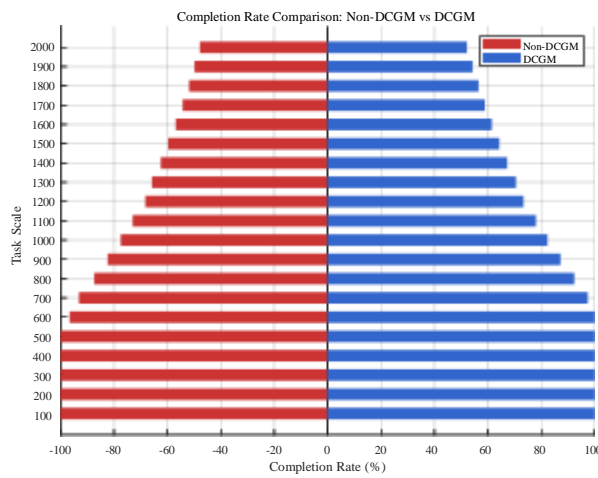


Fig. 10 Task completion rate comparison: DCGM vs. non-DCGM.

The above experimental data confirm that the proposed DCGM effectively decomposes large-scale AEOSSP into multiple smaller-scale subproblems, thereby significantly reducing computational costs and improving scheduling quality.

4.5 Effectiveness Verification of the Adaptive Reward-Penalty Search Strategy

Traditional GWO algorithms primarily rely on fixed-position update mechanisms, which may limit their adaptability in complex or large-scale scheduling scenarios. To evaluate the effectiveness of the proposed ARPS strategy, this section conducts a comparative analysis against the conventional GWO search mechanism under the same experimental settings described in Sections 4.1 and 4.2.

Table 6 Comparison of Solution Performance under GWO's strategy and ARPS strategy

task scale	Initial Solution	GWO's strategy		ARPS	
		Iterations	Optimal Solution	Iterations	Optimal Solution
100	100	1	100	1	100
200	200	1	200	1	200
300	300	1	300	1	300
400	400	1	400	1	400
500	500	10	497.6	1	500
600	591.4	16	598.2	7	600
700	662.9	20	670.7	11	681.9
800	724.8	30	726.1	16	739.1
900	762.4	37	768.5	23	784.8
1000	804.1	40	804.1	28	823.3
1100	835.9	48	844.4	33	858.2
1200	858.7	53	874.5	39	880.3
1300	901.4	55	901.4	46	917.0
1400	905.7	63	911.8	52	941.4
1500	923.5	70	945.3	58	964.5
1600	958.2	75	958.4	64	983.8
1700	965.1	80	970.6	70	1000.6
1800	978.4	82	990.5	77	1017.4
1900	981.3	90	981.3	83	1031.3
2000	984.6	98	984.6	89	1043.4

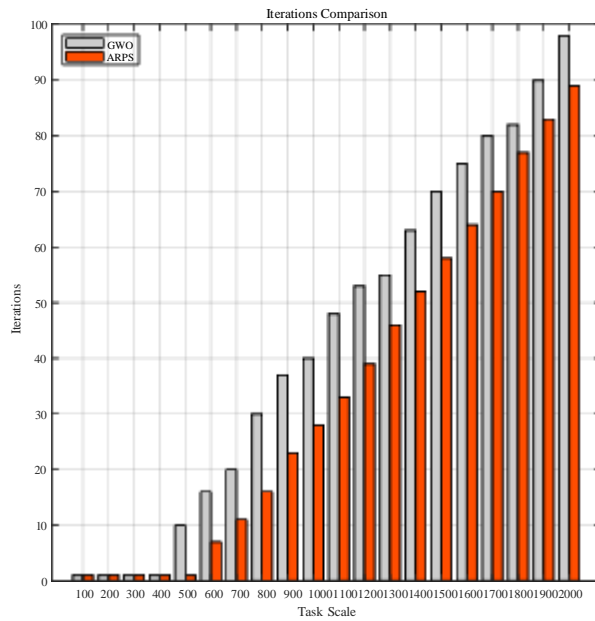


Fig. 11 Iterations comparison: ARPS vs. GWO's strategy.

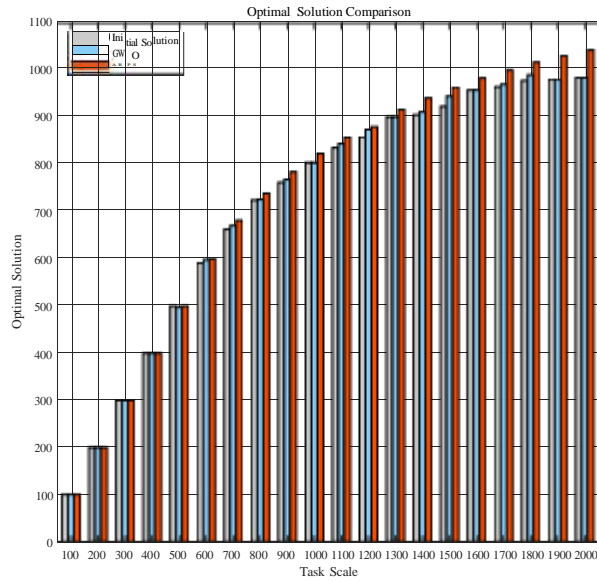


Fig. 12 Solution quality comparison: APRS vs. GWO's strategy.

The results presented in Table 6 quantitatively compare the performance of the ARPS and standard GWO strategies in terms of iteration efficiency and solution quality. Fig. 11 illustrates the number of iterations required to reach the optimal solution across different task scales. For small-scale problems (i.e., below 500 tasks), both ARPS and GWO exhibit rapid convergence with negligible differences in iteration counts. However, as the task scale increases, the iteration count for the standard GWO strategy grows significantly faster than that for ARPS. For example, at the 2000-task level, GWO requires 98 iterations to reach convergence, while ARPS achieves convergence in only 89 iterations, representing a relative reduction of approximately 9.2%. This consistent reduction in iteration counts across large-scale instances demonstrates the efficiency gains provided by the adaptive reward-penalty mechanism, which dynamically adjusts search direction and intensity based on real-time feedback, effectively enhancing both search speed and solution quality.

Fig.12 reveals some stagnation in GWO's performance at certain task scales. For instance, at task scales of 1000, 1900, and 2000, the optimal solutions found by GWO remain unchanged from the initial solution, indicating that the algorithm failed to generate any effective improvement during the optimization process.

This phenomenon can be attributed to the inherent design of the GWO algorithm, which relies on α , β , and δ wolves to guide the search process. In the context of the AEOSSP, these three types of wolves correspond to solutions of descending quality levels. Since AEOSSP is a highly discrete optimization problem, the quality gap among these solution levels can be substantial, which in turn weakens the effectiveness of the β and δ wolves in providing useful search directions, particularly when the α wolf is already positioned in a locally optimal region. Moreover, the lack of a quantitative task priority mechanism in GWO further limits its capacity to escape local optima,

particularly in large-scale instances where the solution space is highly rugged and complex.

In contrast, the ARPS exhibits consistent improvements across all task scales, avoiding the stagnation seen in GWO. This is primarily due to its focused exploitation under the guidance of the head wolf, combined with the ARPS strategy, which quantitatively prioritizes tasks during the optimization process. As a result, ARPS is better equipped to escape local optima and drive the search toward higher-quality solutions, which is clearly reflected in the superior solution quality observed for large-scale tasks.

5 CONCLUSION

This paper presents a novel Grey Wolf Optimizer (GWO) specifically designed for the AEOSSP, incorporating several key innovations to address the unique challenges of large-scale satellite mission planning. The proposed algorithm introduces a Dynamic Conflict Grouping Method (DCGM) to partition continuous observation arcs into multiple independent wolf packs, effectively reducing solution space dimensionality and enhancing computational efficiency. This approach allows the algorithm to balance intra-group conflict resolution and inter-group independence, significantly improving scalability. To further enhance solution quality, a Dual-Time Heuristic Initial Solution (DTHS) strategy is developed, integrating deadline urgency assessment and dynamic start-time scoring. This component ensures the generation of high-quality initial solutions, reducing the risk of premature convergence and accelerating the overall search process. Additionally, the algorithm incorporates an Adaptive Reward-Penalty Search (ARPS) strategy, which dynamically adjusts search intensity based on real-time feedback from the leading wolves, guiding the population toward optimal solutions. This approach not only improves local exploitation but also mitigates the risk of becoming trapped in local optima. Extensive experimental results demonstrate that the proposed GWO algorithm consistently outperforms both state-of-the-art (SOTA) and classical baseline methods in terms of computational efficiency and solution quality. The advantages of the GWO become increasingly pronounced as the problem scale grows, highlighting its potential for large-scale AEOSSP and real-time satellite mission planning.

However, several limitations remain. First, the current DCGM approach primarily focuses on reducing conflict within predefined wolf packs, potentially overlooking more dynamic conflict resolution strategies that could further improve scalability. Second, the DTHS strategy, while effective in reducing initial search time, may still struggle with highly dynamic and unpredictable task environments. Third, the ARPS strategy relies on pre-set reward-penalty parameters, which may require manual tuning for different problem instances. Future work will focus on addressing these limitations by exploring adaptive conflict grouping methods, integrating machine learning techniques for dynamic parameter tuning, and incorporating real-time environmental feedback to further enhance the robustness and flexibility of the GWO algorithm. Additionally, the potential integration of multi-objective optimization strategies will be investigated to extend the applicability of the algorithm to more complex satellite mission planning scenarios.

6 Declarations

Data Availability

The data used in this study involve military-sensitive information and are not publicly available. Interested researchers may contact the corresponding author to discuss potential access under appropriate confidentiality agreements.

Author Contributions

Conceptualization: Chen Wang. (lead); Pengfei Gao. (supporting).

Data curation: Chen Wang, Kangning Du.

Formal analysis: Chen Wang, Kangning Du.

Methodology: Chen Wang, Kangning Du, Pengfei Gao.

Investigation: Chen Wang, Kangning Du.

Writing – original draft: Chen Wang.

Writing – review & editing: Chen Wang, Kangning Du, Pengfei Gao.

Supervision: Lin Cao.

Funding acquisition: Lin Cao.

Declaration of Interest Statement

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Ethics approval and informed consent

The data used in this study are from a sensitive industry project and have passed institutional ethical review. Due to confidentiality, the data are not publicly available. For academic collaboration, please contact the corresponding author.

References

- [1] Du, J., Jiang, C., Guo, Q., Guizani, M., Ren, Y.: Cooperative earth observation through complex space information networks. *IEEE Wireless Commun.* **23**(2), 136–144 (2016) <https://doi.org/10.1109/MWC.2016.7462495>
- [2] Chen, H., Peng, S., Du, C., Li, J.: Description and analysis of the eos task scheduling problem. *Earth Observation Satellites*, 2 (2023) https://doi.org/10.1007/978-981-99-3565-9_2
- [3] Wang, X., Wu, G., Xing, L., Pedrycz, W.: Agile earth observation satellite scheduling over 20 years: Formulations, methods, and future directions. *IEEE Syst. J.* **15**(3), 3881–3892 (2021) <https://doi.org/10.1109/JSYST.2020.2997050>
- [4] Globus, A., Crawford, J., Lohn, J., Pryor, A.: A comparison of techniques for scheduling earth observing satellites. In: *Proc. 19th Natl. Conf. Artif. Intell. (AAAI)*, San Jose, CA, USA, pp. 836–843 (2004)

- [5] Wang, X., Leus, R., Han, C.: Fixed interval scheduling of multiple earth observation satellites with multiple observations. In: Proc. 9th Int. Conf. Mech. Aerosp. Eng. (ICMAE), Budapest, Hungary, pp. 28–33 (2018)
- [6] Bianchessi, N., Cordeau, J.-F., Desrosiers, J., Laporte, G., Raymond, V.: A heuristic for the multi-satellite, multi-orbit and multi-user management of earth observation satellites. *Eur. J. Oper. Res.* **177**(2), 750–762 (2007) <https://doi.org/10.1016/j.ejor.2005.12.026>
- [7] Valicka, C.G., Garcia, D., Staid, A., Watson, J.-P., Hackebeil, G., Rathinam, S., Ntamo, L.: Mixed-integer programming models for optimal constellation scheduling given cloud cover uncertainty. *Eur. J. Oper. Res.* **275**(2), 431–445 (2019) <https://doi.org/10.1016/j.ejor.2018.11.043>
- [8] Chu, X., Chen, Y., Xing, L.: A branch and bound algorithm for agile earth observation satellite scheduling. *Discrete Dyn. Nat. Soc.* **2017**, 1–15 (2017) <https://doi.org/10.1155/2017/7345941>
- [9] Chu, X., Chen, Y., Tan, Y.: An anytime branch and bound algorithm for agile earth observation satellite onboard scheduling. *Adv. Sp. Res.* **60**(9), 2077–2090 (2017) <https://doi.org/10.1016/j.asr.2017.07.026>
- [10] Lemaître, M., Verfaillie, G., Jouhaud, F., Lachiver, J.-M., Bataille, N.: Selecting and scheduling observations of agile satellites. *Aerosp. Sci. Technol.* **6**(5), 367–381 (2002) [https://doi.org/10.1016/S1270-9638\(02\)01173-2](https://doi.org/10.1016/S1270-9638(02)01173-2)
- [11] Verfaillie, G., Lemaître, M.: Selecting and scheduling observations for agile satellites: Some lessons from the constraint reasoning community point of view, pp. 670–684. Springer, Berlin, Germany (2001). https://doi.org/10.1007/3-540-45578-7_45
- [12] Lematre, M., Verfaillie, G., Fargier, H., Lang, J., Bataille, N., *et al.*: Sharing the use of earth observation satellites. In: Proc. 3rd Workshop on Planning and Scheduling for Space, Houston, TX, USA (2002). [Online]. Available: <https://hal.science/hal-04269869>
- [13] Wolfe, W.J., Sorensen, S.E.: Three scheduling algorithms applied to the earth observing systems domain. *Manage. Sci.* **46**(2), 148–166 (2000) <https://doi.org/10.1287/MNSC.46.1.148.15134>
- [14] Wang, X.-W., Chen, Z., Han, C.: Scheduling for single agile satellite, redundant targets problem using complex networks theory. *Chaos Solit. Fract.* **83**, 125–132 (2016) <https://doi.org/10.1016/j.chaos.2015.12.003>
- [15] Wang, X., Han, C., Zhang, R., Gu, Y.: Scheduling multiple agile earth observation satellites for oversubscribed targets using complex networks theory. *IEEE Access* **7**, 110605–110615 (2019) <https://doi.org/10.1109/ACCESS.2019.2925704>

- [16] Liu, X., Laporte, G., Chen, Y., He, R.: An adaptive large neighborhood search metaheuristic for agile satellite scheduling with time-dependent transition time. *Comput. Oper. Res.* **86**, 41–53 (2017) <https://doi.org/10.1016/j.cor.2017.04.006>
- [17] Peng, G., Dewil, R., Verbeeck, C., Gunawan, A., Xing, L., Vansteenwegen, P.: Agile earth observation satellite scheduling: An orienteering problem with time-dependent profits and travel times. *Comput. Oper. Res.* **111**, 84–98 (2019) <https://doi.org/10.1016/j.cor.2019.05.030>
- [18] He, L., Liu, X., Laporte, G., Chen, Y., Chen, Y.: An improved adaptive large neighborhood search algorithm for multiple agile satellites scheduling. *Comput. Oper. Res.* **100**, 12–25 (2018) <https://doi.org/10.1016/j.cor.2018.06.020>
- [19] Chang, Z., Zhou, Z.: Three multi-objective memetic algorithms for observation scheduling problem of active-imaging agile earth observation satellites. *Ann. Oper. Res.* **346**, 861–893 (2025) <https://doi.org/10.1007/s10479-024-06156-5>
- [20] Peng, G., Song, G., He, Y., Yu, J., Xiang, S., Xing, L.: Solving the agile earth observation satellite scheduling problem with time-dependent transition times. *IEEE Trans. Syst., Man, Cybern.: Syst.* **52**(3), 1614–1625 (2022) <https://doi.org/10.1109/TSMC.2020.3031738>
- [21] Mirjalili, S., Mirjalili, S.M., Lewis, A.: Grey wolf optimizer. *Adv. Eng. Softw.* **69**, 46–61 (2014) <https://doi.org/10.1016/j.advengsoft.2013.12.007>