Implicit Q-learning-bootstrapped ant colony optimization for maritime moving-target observation scheduling with agile satellites

He Wang[a], Junyu Wu[a], Yeye Liu[b], Yifan Zhou[a], Jie Zhang[a], Hui Li[a], Yanjie Song[c], Liang Li[a,*]

[a] *College of Intelligent Science and Engineering, Harbin Engineering University, Harbin 150001, China*
[b] *School of Electrical and Control Engineering, North University of China, Taiyuan 030051, China*
[c] *School of Information Science and Technology, Dalian Maritime University, Dalian 116026, China*

arXiv:2608.24471v1 [cs.AI] 25 Aug 2026

Abstract

Maritime moving-target observation scheduling with agile Earth observation satellites is a dynamic, sequence-dependent combinatorial optimization problem. Sea-surface targets move continuously, causing feasible observation windows to vary with target motion and satellite orbital geometry. The scheduler must jointly determine task selection, satellite assignment, observation-window selection, and observation ordering under time-window, attitude-maneuvering, onboard-resource, and cloud-affected availability constraints. This paper proposes an implicit Q-learning-bootstrapped ant colony optimization method, termed IQACO, for multi-satellite maritime moving-target observation scheduling. Rather than directly learning a task-selection policy, IQACO embeds an offline implicit Q-learning module into constructive ant colony optimization to adaptively adjust the pheromone factor, heuristic factor, and evaporation rate. A compact search-state representation captures pheromone distribution, current and historical-best solution quality, and iteration progress. During online scheduling, ant colony optimization constructs feasible observation sequences, while the learned policy regulates exploration and exploitation according to the current search state. Experiments on 14 scenarios with different scales and satellite configurations show that IQACO obtains the highest mean observation benefit in every scenario, improves the result of conventional ant colony optimization by 3.40%–9.40%, accelerates convergence, and remains stable under different objective-weight settings. These results demonstrate that offline value learning provides an effective adaptive search-control mechanism for constrained maritime moving-target observation scheduling.

Keywords: Agile satellite scheduling, Maritime moving target, Ant colony optimization, Offline reinforcement learning, Implicit Q-learning

*Corresponding author.
Email addresses: wang_he@hrbeu.edu.cn (He Wang), wujunyu@hrbeu.edu.cn (Junyu Wu), 20240042@nuc.edu.cn (Yeye Liu), ercuniociao@163.com (Yifan Zhou), jiezhang@hrbeu.edu.cn (Jie Zhang), lihuiheu@hrbeu.edu.cn (Hui Li), songyj_2017@163.com (Yanjie Song), liliang@hrbeu.edu.cn (Liang Li)

1. Introduction

Maritime moving-target observation using agile Earth observation satellites (AEOSs) is important for vessel traffic monitoring, maritime search and rescue, illegal fishing detection, maritime security, and environmental surveillance. Compared with conventional satellites with limited pointing capability, AEOSs can rapidly slew their payloads toward sea-surface targets within limited visibility intervals. When multiple satellites are coordinated to observe numerous moving targets, the scheduling problem is no longer a simple target-selection problem, but a coupled decision-making problem involving satellite-task assignment, observation-window selection, observation sequencing, and feasibility checking under attitude-maneuvering and resource constraints [1, 2, 3]. The overall scheduling scenario is illustrated in Fig. 1.

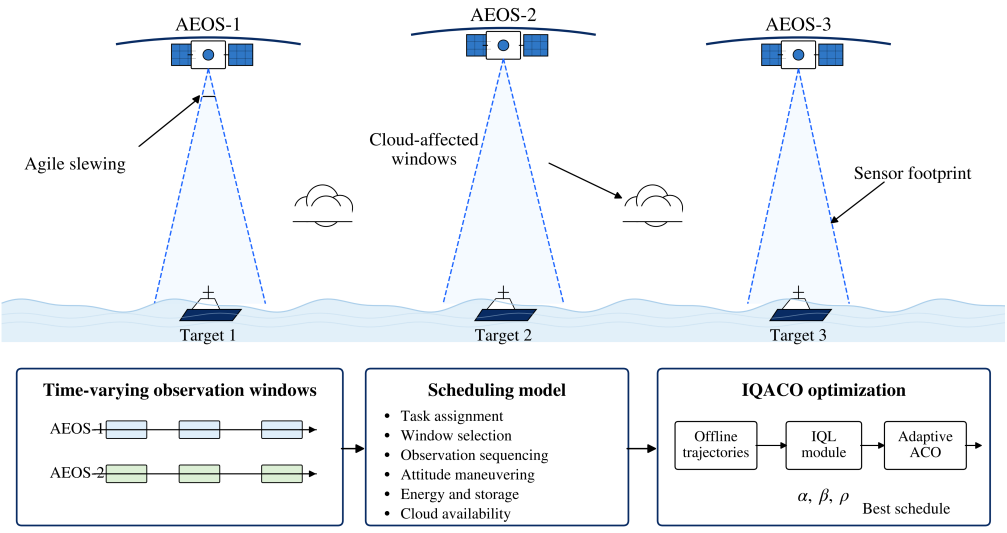

Figure 1: Schematic illustration of multi-satellite maritime moving target scheduling with time-varying observation windows, operational constraints, and IQL-guided adaptive ACO optimization.

Maritime moving-target scheduling is more challenging than static-target scheduling. Sea-surface targets continuously change their relative geometry with satellite orbits and sensor footprints, making feasible observation windows strongly time dependent. In addition, schedule feasibility depends on the execution order of selected tasks because attitude-maneuvering time and maneuvering energy are required between consecutive observations. Frequent slews and high-resolution imaging further consume limited onboard energy and storage resources [4, 5, 6, 7]. For optical payloads, ocean cloud cover may reduce the practical value of geometrically visible windows [8, 9, 10]. These factors jointly lead to a large-scale, sequence-dependent, and resource-constrained combinatorial optimization problem.

AEOS scheduling has been studied for more than two decades. Existing methods include exact optimization, heuristic and local-search methods, metaheuristics, and learning-based approaches [11]. Exact methods, such as integer programming, branch-and-bound, column generation, and constraint programming, can provide rigorous solutions for small or medium instances, but their computational cost increases rapidly when sequence-dependent transitions, onboard resources, and uncertainty are jointly considered [2, 12, 13]. Heuristic and

local-search methods are efficient and easy to implement, but their performance often depends on hand-crafted rules and problem-specific assumptions [14, 15]. Metaheuristics, such as GA, PSO, WOA, ACO, memetic algorithms, and hybrid neighborhood search, have therefore been widely used for large-scale AEOS scheduling [1, 16, 17, 18, 19].

Among metaheuristics, ACO is naturally suitable for sequence construction because pheromone information and heuristic information can guide task-transition decisions. However, conventional ACO usually relies on fixed parameters, including the pheromone factor α, heuristic factor β, and evaporation rate ρ [20]. Fixed parameters may not maintain an appropriate exploration–exploitation balance when target density, satellite number, feasible-window distribution, and resource constraints change across scenarios. Therefore, an adaptive parameter-control mechanism is needed to adjust the search behavior according to the current optimization state.

Reinforcement learning (RL) has recently attracted attention in AEOS scheduling because it can learn scheduling policies or value functions from data [21, 22, 23, 24, 25, 26]. However, many existing RL-based schedulers directly learn task-selection or sequence-generation policies. Such methods often face large discrete action spaces, dynamically changing feasible action sets, and limited generalization to scenarios with different target-window distributions. Moreover, stand-alone RL schedulers must explicitly handle hard operational constraints during every decision step, which increases the learning burden and may reduce feasibility reliability.

Offline RL provides another possible way to introduce learning into satellite scheduling because it learns from collected decision trajectories without costly online trial-and-error [27, 28]. Implicit Q-learning (IQL) is particularly suitable for offline learning because it avoids explicit evaluation of out-of-distribution actions [29]. Nevertheless, directly using IQL as a stand-alone scheduler remains difficult for discrete, constraint-intensive, and sequence-dependent AEOS scheduling. A more practical strategy is to use IQL as a value-guided parameter controller within a constructive metaheuristic. In this way, the learning module adapts the search behavior, while the deterministic decoder remains responsible for feasible schedule construction.

To address these issues, this article develops an implicit Q-learning-bootstrapped ant colony optimization method, termed IQACO, for multi-satellite maritime moving-target observation scheduling. IQACO uses an offline-trained IQL policy to adaptively adjust α, β, and ρ according to a compact five-dimensional search state. ACO remains responsible for constructing feasible schedules under time-window, attitude-maneuvering, energy, storage, and cloud-affected availability constraints. This design preserves the feasibility-oriented search capability of ACO while introducing value-guided exploration–exploitation control.

The main contributions of this article are summarized as follows:

1. A multi-satellite maritime moving-target observation scheduling model is formulated by integrating time-varying observation windows, satellite-task-window assignment, sequence-dependent attitude maneuvering, onboard energy and storage resources, and cloud-affected availability.
2. An IQACO method is developed to adaptively regulate ACO parameters using offline IQL. The learned policy adjusts the pheromone factor, heuristic factor, and evaporation rate according to the current search state, improving exploration–exploitation balance during feasible schedule construction.

3. Comprehensive experiments are conducted on 14 scenarios with different problem scales and satellite configurations. The results verify the convergence performance, final observation benefit, objective-weight sensitivity, and training behavior of the proposed method.

The remainder of this article is organized as follows. Section 2 formulates the scheduling model. Section 3 presents the proposed IQACO method. Section 4 reports the simulation experiments and result analysis. Section 5 concludes this article.

2. Scheduling Model

2.1. Problem Description

We consider multi-satellite maritime moving-target observation scheduling over a finite planning horizon T_H. The scheduling objects include a set of agile satellites and a set of maritime moving targets. Before optimization, target trajectories and satellite ephemerides are propagated to generate feasible task–satellite observation windows. The scheduler then determines task selection, satellite assignment, observation-window selection, and task ordering on each satellite under time-window, attitude-maneuvering, onboard energy, storage-capacity, and cloud-affected availability constraints.

Compared with static-target scheduling, maritime moving-target observation has stronger temporal and spatial variability. The relative geometry between satellites and targets changes with both vessel motion and orbital motion, making feasible observation windows time dependent. A geometrically visible window may also have low practical availability due to ocean cloud cover. For agile satellites, schedule feasibility further depends on the execution order of selected tasks because attitude-maneuvering time and energy are required between consecutive observations. Therefore, the problem is a sequence-dependent and resource-constrained combinatorial optimization problem.

2.2. Sets and Parameters

The main notation used in the scheduling model is listed in Table 1. An original moving target may have multiple observation requirements within the planning horizon; these requirements are expanded into independent scheduling tasks. If task i cannot be observed by satellite s, the corresponding feasible-window set $\mathcal{W}_{i,s}$ is empty.

Two binary variables are used to describe the scheduling decision. The variable $x_{i,s,k}$ indicates whether task i is assigned to satellite s and executed in window k. It is set to one if the corresponding task–satellite–window combination is selected, and zero otherwise:

$$x_{i,s,k} = \begin{cases} 1, & \text{selected}, \\ 0, & \text{otherwise}. \end{cases} \tag{1}$$

The variable y_{ij}^{s} describes the immediate-successor relation in the task sequence of satellite s. It is set to one if satellite s executes task j immediately after task i, and zero otherwise:

$$y_{ij}^{s} = \begin{cases} 1, & \text{if } j \text{ immediately follows } i, \\ 0, & \text{otherwise}. \end{cases} \tag{2}$$

Thus, $x_{i,s,k}$ determines task selection, satellite assignment, and observation-window selection, while y_{ij}^{s} describes the task order on each satellite. The latter is also used to compute the attitude-maneuvering time and energy between consecutive observations.

Table 1: Notation used in the scheduling model

Symbol	Description
$\mathcal{T}$, $\mathcal{S}$	Sets of scheduling tasks and agile satellites
$\mathcal{W}_{i,s}$	Feasible observation-window set for task i and satellite s
i, j	Task indices
s	Satellite index
k, l	Observation-window indices
N_T, N_S	Numbers of tasks and satellites
T_H	Planning-horizon length
$t_{i,s,k}^{\mathrm{start}}$	Start time of window k
$t_{i,s,k}^{\mathrm{end}}$	End time of window k
$t_{i,s,k}^{\mathrm{obs}}$	Observation start time
d_i, p_i	Observation duration and nominal benefit of task i
$c_{i,s,k}$, $c_{\min}$	Cloud-affected availability factor and threshold
θ_{ij}^{s}	Slewing angle from task i to task j
$T_{ij}^{s,\mathrm{man}}$	Maneuvering time from task i to task j
$\omega_s^{\max}$	Maximum angular rate of satellite s
$a_s^{\max}$	Maximum angular acceleration of satellite s
$e_{i,s,k}^{\mathrm{obs}}$	Imaging energy
$e_{ij}^{s,\mathrm{man}}$	Maneuvering energy
$E_s^{\max}$, E_s^{use}	Energy capacity and energy use of satellite s
$m_{i,s,k}$, $M_s^{\max}$	Generated data volume and storage capacity
$x_{i,s,k}$	Task-window assignment variable
y_{ij}^{s}	Immediate-successor variable

2.3. Constraints

A feasible schedule must satisfy operational constraints related to task assignment, observation windows, task sequencing, attitude maneuvering, onboard resources, and cloud-affected availability. These constraints are used as feasibility rules in the constructive decoder.

2.3.1. Task Uniqueness Constraint

Each task is executed at most once:

$$\sum_{s\in\mathcal{S}}\sum_{k\in\mathcal{W}_{i,s}} x_{i,s,k} \le 1, \quad \forall i \in \mathcal{T}. \tag{3}$$

2.3.2. Time-Window Feasibility Constraint

The observation interval must be contained within the selected window:

$$t_{i,s,k}^{\mathrm{start}} \le t_{i,s,k}^{\mathrm{obs}}, \tag{4a}$$

$$t_{i,s,k}^{\mathrm{obs}} + d_i \le t_{i,s,k}^{\mathrm{end}}. \tag{4b}$$

These inequalities are enforced only for selected task–satellite–window nodes with $x_{i,s,k} = 1$.

2.3.3. Sequence-Linking Constraint

The sequence variable y_{ij}^s is valid only when both tasks i and j are assigned to satellite s:

$$y_{ij}^s \leq \sum_{k \in \mathcal{W}_{i,s}} x_{i,s,k}, \qquad \forall i, j \in \mathcal{T},\ i \neq j,\ s \in \mathcal{S}, \tag{5}$$

$$y_{ij}^s \leq \sum_{l \in \mathcal{W}_{j,s}} x_{j,s,l}, \qquad \forall i, j \in \mathcal{T},\ i \neq j,\ s \in \mathcal{S}. \tag{6}$$

Each scheduled task has at most one immediate successor and predecessor:

$$\sum_{\substack{j \in \mathcal{T} \\ j \neq i}} y_{ij}^s \leq \sum_{k \in \mathcal{W}_{i,s}} x_{i,s,k}, \qquad \forall i \in \mathcal{T},\ s \in \mathcal{S}, \tag{7}$$

$$\sum_{\substack{i \in \mathcal{T} \\ i \neq j}} y_{ij}^s \leq \sum_{l \in \mathcal{W}_{j,s}} x_{j,s,l}, \qquad \forall j \in \mathcal{T},\ s \in \mathcal{S}. \tag{8}$$

These constraints define the consistency between task assignment and immediate-successor relations. The actual satellite-specific task sequences are generated explicitly by the constructive ACO procedure, where successor relations, temporal feasibility, and resource feasibility are checked during schedule construction.

2.3.4. Attitude Maneuvering Constraint

For agile satellites, an attitude maneuver is required between two consecutive observations assigned to the same satellite. The required slewing angle θ_{ij}^s is computed from the angular separation between the payload pointing directions of tasks i and j. A rate- and acceleration-limited maneuvering model is used, as illustrated in Fig. 2. If the required angle is large enough, the satellite reaches the maximum angular rate and follows a trapezoidal profile; otherwise, it follows a triangular profile.

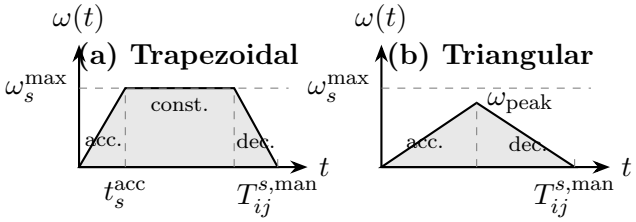

Figure 2: Attitude-maneuvering angular-rate profiles.

Let $\omega_s^{\max}$ and $a_s^{\max}$ denote the maximum angular rate and maximum angular acceleration of satellite s, respectively. The acceleration or deceleration time is

$$t_s^{\text{acc}} = \frac{\omega_s^{\max}}{a_s^{\max}}. \tag{9}$$

When the satellite can reach the maximum angular rate, the maneuvering time is

$$T_{ij}^{s,\text{man}} = 2t_s^{\text{acc}} + \frac{\theta_{ij}^s - (\omega_s^{\max})^2/a_s^{\max}}{\omega_s^{\max}}, \tag{10}$$

where $\theta_{ij}^{s} \geq (\omega_s^{\max})^2/a_s^{\max}$. When the maximum angular rate cannot be reached, the maneuvering time is

$$T_{ij}^{s,\mathrm{man}} = 2\sqrt{\frac{\theta_{ij}^{s}}{a_s^{\max}}}, \tag{11}$$

where $\theta_{ij}^{s} < (\omega_s^{\max})^2/a_s^{\max}$.

When task j immediately follows task i on satellite s and their selected windows are k and l, respectively, the observation start times must satisfy

$$t_{j,s,l}^{\mathrm{obs}} \geq t_{i,s,k}^{\mathrm{obs}} + d_i + T_{ij}^{s,\mathrm{man}}. \tag{12}$$

This constraint reserves sufficient transition time between consecutive observations on the same satellite.

2.3.5. Onboard Energy Constraint

The total energy use, including imaging and maneuvering energy, must not exceed the onboard capacity:

$$\sum_{i\in\mathcal{T}} \sum_{k\in\mathcal{W}_{i,s}} e_{i,s,k}^{\mathrm{obs}} x_{i,s,k} + \sum_{\substack{i,j\in\mathcal{T} \\ i\neq j}} e_{ij}^{s,\mathrm{man}} y_{ij}^{s} \leq E_s^{\max}, \quad \forall s \in \mathcal{S}. \tag{13}$$

2.3.6. Onboard Storage Constraint

The generated data volume must not exceed the storage capacity:

$$\sum_{i\in\mathcal{T}} \sum_{k\in\mathcal{W}_{i,s}} m_{i,s,k} x_{i,s,k} \leq M_s^{\max}, \quad \forall s \in \mathcal{S}. \tag{14}$$

2.3.7. Cloud-Availability Constraint

For optical observations, a geometrically visible window may still have low practical value due to cloud cover. Let $c_{i,s,k}$ denote the cloud-affected availability of executing task i by satellite s in window k. Candidate windows with availability lower than the minimum acceptable threshold $c_{\min}$ are excluded:

$$x_{i,s,k} = 0, \quad \text{if} \quad c_{i,s,k} < c_{\min}. \tag{15}$$

For retained windows, $c_{i,s,k}$ is further used as a benefit attenuation factor in the objective function.

2.4. Objective Function

Under the above feasibility constraints, the objective is to maximize the overall observation performance of the satellite constellation. Three normalized components are considered: observation benefit, energy efficiency, and workload balance. Normalization allows the weights to express mission preferences rather than compensate for different physical scales. The objective function is formulated as

$$\max F = \eta_1 F_p + \eta_2 F_e + \eta_3 F_b, \tag{16}$$

where F_p, F_e, and F_b denote the normalized observation-benefit, energy-efficiency, and workload-balance terms, respectively. The objective weights satisfy

$$\eta_1 + \eta_2 + \eta_3 = 1, \qquad \eta_1, \eta_2, \eta_3 \geq 0. \tag{17}$$

The normalized observation-benefit term is defined as

$$F_p = \frac{\sum_{i\in\mathcal{T}} \sum_{s\in\mathcal{S}} \sum_{k\in\mathcal{W}_{i,s}} p_i c_{i,s,k} x_{i,s,k}}{\sum_{i\in\mathcal{T}} p_i}. \tag{18}$$

This term measures the effective benefit obtained from the selected tasks. The factor $c_{i,s,k}$ reduces the benefit of a window with low cloud-affected availability.

For satellite s, the energy use is composed of imaging energy and attitude-maneuvering energy:

$$E_s^{\text{use}} = \sum_{i\in\mathcal{T}} \sum_{k\in\mathcal{W}_{i,s}} e_{i,s,k}^{\text{obs}} x_{i,s,k} + \sum_{\substack{i,j\in\mathcal{T} \\ i\neq j}} e_{ij}^{s,\text{man}} y_{ij}^{s}. \tag{19}$$

The energy-efficiency term is then defined as

$$F_e = 1 - \frac{\sum_{s\in\mathcal{S}} E_s^{\text{use}}}{\sum_{s\in\mathcal{S}} E_s^{\max}}. \tag{20}$$

A larger F_e indicates lower relative energy use.

The workload of satellite s is defined as the total duration of its selected observations:

$$L_s = \sum_{i\in\mathcal{T}} \sum_{k\in\mathcal{W}_{i,s}} d_i x_{i,s,k}. \tag{21}$$

The average workload of the constellation is

$$\bar{L} = \frac{1}{N_S} \sum_{s\in\mathcal{S}} L_s. \tag{22}$$

The workload-balance term is defined as

$$F_b = \left(1 + \frac{\sqrt{\frac{1}{N_S} \sum_{s\in\mathcal{S}} (L_s - \bar{L})^2}}{\bar{L} + \epsilon} \right)^{-1}, \tag{23}$$

where ϵ is a small positive constant used to avoid division by zero. A larger F_b indicates a more balanced workload distribution among satellites.

Therefore, the objective favors high-benefit and practically available observations while reducing relative energy consumption and avoiding excessive workload concentration on a small number of satellites.

3. Method

3.1. Method Overview

IQACO consists of an offline IQL training stage and an online ACO scheduling stage, as shown in Fig. 3. In the offline stage, ACO is executed on training scenarios with exploratory parameter adjustments, and the resulting transitions $(\mathbf{s}_t, \mathbf{a}_t, R_t, \mathbf{s}_{t+1}, d_t)$ are collected to train an IQL policy. In the online stage, ACO constructs feasible schedules at each iteration, updates the best solution and pheromone matrix, and then uses the trained policy to adjust α, β, and ρ according to the current search state.

Unlike direct RL schedulers that output discrete task-selection actions, IQACO only learns continuous parameter adjustments for ACO. Feasible schedule construction is still handled by the deterministic decoder, which reduces the learning burden and improves constraint satisfaction. Thus, IQL is used as a value-guided search controller rather than a replacement for the scheduling algorithm.

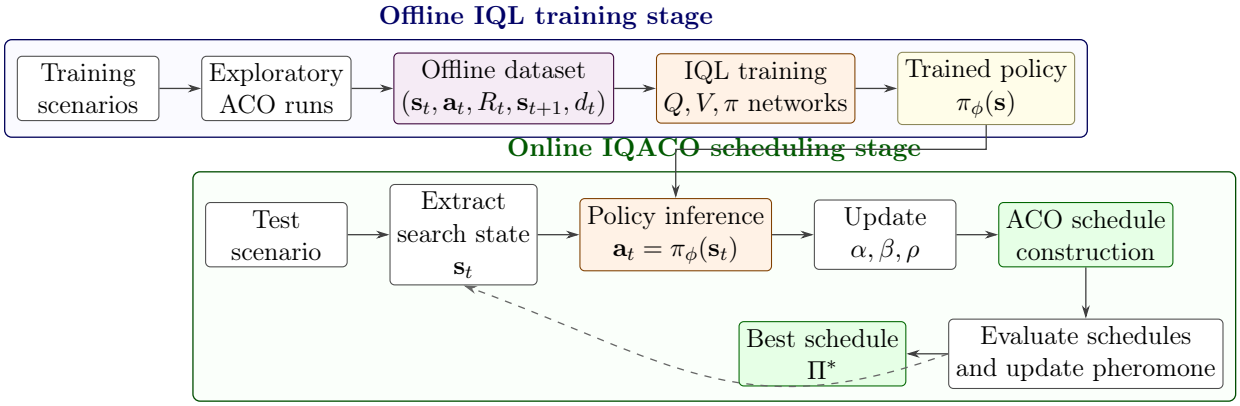

Figure 3: Workflow of the proposed IQACO method.

3.2. Solution Encoding and ACO-Based Schedule Construction

A candidate observation is represented by a task–satellite–window node $z = (i, s, k)$, where i, s, and k denote the task, satellite, and feasible observation window, respectively. Candidate nodes are pre-filtered according to visibility, time-window feasibility, and cloud-affected availability. A complete schedule is an ordered node list $\Pi = \{z_1, z_2, \ldots, z_{|\Pi|}\}$, which can be decomposed into satellite-specific sequences $\Pi = \{\Pi_1, \Pi_2, \ldots, \Pi_{N_S}\}$, as shown in Fig. 4. The upper level of the representation stores the complete multi-satellite schedule, whereas the lower level preserves the chronological node sequence assigned to each satellite. Because each node explicitly records the selected task, satellite, and observation window, predecessor–successor relations can be recovered directly for maneuver-time, maneuver-energy, temporal-feasibility, and resource-feasibility checks. This hierarchical encoding therefore captures task selection, satellite assignment, window selection, and observation ordering within a single constructive representation.

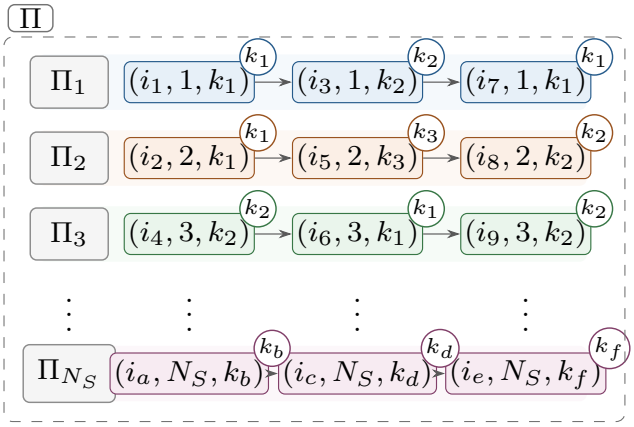

Figure 4: Hierarchical solution encoding. The global schedule is decomposed into chronological satellite-specific sequences of task–satellite–window nodes.

Each ant incrementally selects schedulable tasks from a feasible candidate set $\mathcal{C}^m$ that excludes candidates violating operational constraints. Algorithm 1 summarizes this construction. The evaporation rate ρ is applied only during global pheromone updating, not during single-ant construction.

For the mth ant, the next candidate node $z = (i_z, s_z, k_z) \in \mathcal{C}^m$ is selected with probability

$$P_m(z|u) = \frac{[\tau_{u,i_z}]^\alpha [\eta_m(u,z)]^\beta}{\sum_{q\in\mathcal{C}^m} [\tau_{u_q,i_q}]^\alpha [\eta_m(u_q,q)]^\beta}, \tag{24}$$

where u is the latest scheduled task on satellite s_z, τ_{u,i_z} is the task-level pheromone intensity from u to candidate task i_z, and $\eta_m(u,z)$ is the heuristic value of node z under the current partial schedule. For another candidate node $q = (i_q, s_q, k_q)$ in the denominator, u_q denotes the latest scheduled task on satellite s_q. The parameters α and β control the relative influence of pheromone information and heuristic information.

The heuristic value combines benefit contribution, energy effect, and workload balance:

$$\eta_m(u,z) = \chi_1 G(u,z) + \chi_2 E_m(u,z) + \chi_3 B_m(z), \tag{25}$$

where $G(u,z)$ denotes the benefit contribution of selecting candidate node z after task u, $E_m(u,z)$ denotes the energy-efficiency contribution considering the additional observation and maneuvering energy, and $B_m(z)$ denotes the workload-balance contribution after inserting z. The coefficients χ_1, χ_2, and χ_3 are nonnegative heuristic weights. These terms are normalized before aggregation so that the heuristic value remains comparable across different scenarios and resource scales.

After all ants have constructed their schedules, each schedule is evaluated by the objective function in Section 2. The pheromone matrix is then updated by evaporation and solution-quality-based deposition:

$$\tau_{ij} \leftarrow \max\{(1-\rho)\tau_{ij}, \tau_{\min}\}, \quad \tau_{ij} \leftarrow \min\{\tau_{ij} + \Delta\tau_{ij}, \tau_{\max}\}, \tag{26}$$

with the deposition increment

$$\Delta\tau_{ij} = \sum_{m=1}^{N_A} \frac{F^m}{F_{\max}} I\big((i,j) \in \Pi^m\big), \tag{27}$$

Algorithm 1 Schedule Construction by One Ant

Require: Candidate observation nodes, pheromone matrix $\boldsymbol{\tau}$, and ACO parameters α and β

Ensure: A feasible schedule Π^m

Initialize $\Pi^m \leftarrow \emptyset$, $\mathcal{U} \leftarrow \mathcal{T}$, and satellite states with virtual initial nodes
while true **do**
    Build the feasible candidate set $\mathcal{C}^m$ from $\mathcal{U}$
    Remove candidates violating time-window, attitude-maneuvering, energy, or storage constraints
    if $\mathcal{C}^m = \emptyset$ **then**
        break
    end if
    Compute the selection probability $P_m(z|u)$ for each $z \in \mathcal{C}^m$
    Select a candidate node $z = (i_z, s_z, k_z)$ by roulette-wheel selection
    Insert z into the satellite-specific sequence $\Pi^m_{s_z}$
    Update the state of satellite s_z, including its latest task and resource states
    Update the unscheduled task set $\mathcal{U} \leftarrow \mathcal{U} \setminus \{i_z\}$
end while
Merge all satellite-specific sequences into Π^m
return Π^m

where N_A is the number of ants, F^m is the objective value of schedule Π^m, $F_{\max}$ is the current best objective, and $I(\cdot)$ is the indicator function.

3.3. Markov Decision Process Formulation

The adaptive control of ACO parameters is formulated as an MDP $\mathcal{M} = (\mathcal{X}, \mathcal{A}, \mathcal{P}, R, \gamma)$, where each decision step corresponds to one ACO iteration. After the tth iteration, the search state is extracted from the pheromone matrix and current scheduling results. The IQL policy then outputs a continuous action to adjust the pheromone factor α, heuristic factor β, and evaporation rate ρ for the next iteration. Here, $\mathcal{X}$ is the state space, $\mathcal{A}$ is the action space, $\mathcal{P}$ is the transition process induced by one ACO iteration, R is the reward function, and γ is the discount factor.

$$\mathbf{s}_t = \left[\bar{\tau}_t,\ \sigma^2_{\tau,t},\ f_t^{\mathrm{cur}},\ f_t^{\mathrm{gb}},\ r_t^{\mathrm{iter}}\right]^T, \tag{28}$$

where $\bar{\tau}_t$ and $\sigma^2_{\tau,t}$ are the normalized mean and variance of the pheromone matrix, respectively. The variables f_t^{cur} and f_t^{gb} denote the best objective value of the current iteration and the global best objective value found so far, respectively. The variable r_t^{iter} is the normalized iteration ratio. These variables describe the pheromone distribution, current search quality, historical best performance, and search progress.

The action is a continuous parameter-adjustment vector:

$$\mathbf{a}_t = [\Delta\alpha_t,\ \Delta\beta_t,\ \Delta\rho_t]^T. \tag{29}$$

After receiving the action, the ACO parameters are updated as

$$\alpha_{t+1} = \mathrm{clip}(\alpha_t + \Delta\alpha_t, \alpha_{\min}, \alpha_{\max}), \tag{30}$$

$$\beta_{t+1} = \mathrm{clip}(\beta_t + \Delta\beta_t, \beta_{\min}, \beta_{\max}), \tag{31}$$

$$\rho_{t+1} = \mathrm{clip}(\rho_t + \Delta\rho_t, \rho_{\min}, \rho_{\max}), \tag{32}$$

where $\mathrm{clip}(\cdot)$ restricts each parameter to its feasible range.

The reward encourages solution improvement while maintaining search diversity. Let f_{t+1}^{cur} be the best objective after applying the adjusted parameters. The improvement term is

$$I_t = \max\left(0,\ f_{t+1}^{\mathrm{cur}} - f_t^{\mathrm{gb}}\right). \tag{33}$$

The final reward is formulated as

$$R_t = \mathrm{clip}\left(c_I I_t + \xi D_t + \kappa B_t,\ R_{\min}, R_{\max}\right), \tag{34}$$

where D_t is a diversity-related term that measures the dispersion of the current search process, and B_t is a bonus term activated when a new global-best solution is obtained. The coefficients c_I, ξ, and κ balance immediate objective improvement, diversity preservation, and global progress. The clipping operation limits excessively large learning targets and improves the stability of offline training.

3.4. Offline IQL Training

The offline dataset is collected by running ACO with exploratory parameter adjustments on training scenarios. During data collection, ACO first performs one iteration to obtain the initial search state. At each subsequent decision step, an exploratory action is sampled to update α, β, and ρ, and the next ACO iteration is executed with the updated parameters. The resulting reward, next state, and terminal indicator are recorded. The dataset is written as

$$\mathcal{D} = \{(\mathbf{s}_t, \mathbf{a}_t, R_t, \mathbf{s}_{t+1}, d_t)\}_{t=1}^{N_D}, \tag{35}$$

where d_t is the terminal indicator and N_D is the number of collected transitions. Algorithm 2 summarizes the transition collection process.

IQL learns a value function $V_\psi(\mathbf{s})$, two Q-functions $Q_{\vartheta_1}(\mathbf{s}, \mathbf{a})$ and $Q_{\vartheta_2}(\mathbf{s}, \mathbf{a})$, and a policy function $\pi_\phi(\mathbf{s})$ from $\mathcal{D}$. The Q-learning target is $y_t = R_t + \gamma(1 - d_t)V_{\bar{\psi}}(\mathbf{s}_{t+1})$, where $V_{\bar{\psi}}$ is the target value network. The Q loss is

$$\mathcal{L}_Q(\vartheta_1, \vartheta_2) = \sum_{j=1}^{2} \mathbb{E}_{\mathcal{D}}\left[\left(Q_{\vartheta_j}(\mathbf{s}_t, \mathbf{a}_t) - y_t\right)^2\right]. \tag{36}$$

The value function uses expectile regression with $\hat{Q} = \min_j Q_{\vartheta_j}$:

$$\begin{aligned} \mathcal{L}_V(\psi) &= \mathbb{E}_{\mathcal{D}}\left[L_{\tau_e}\left(\hat{Q}(\mathbf{s}_t, \mathbf{a}_t) - V_\psi(\mathbf{s}_t)\right)\right], \\ L_{\tau_e}(u) &= |\tau_e - \mathbb{I}(u < 0)|u^2. \end{aligned} \tag{37}$$

The policy is trained by advantage-weighted regression with $A(\mathbf{s}_t, \mathbf{a}_t) = \hat{Q}(\mathbf{s}_t, \mathbf{a}_t) - V_\psi(\mathbf{s}_t)$:

$$\mathcal{L}_\pi(\phi) = \mathbb{E}_{\mathcal{D}}\left[\exp\left(\frac{A(\mathbf{s}_t, \mathbf{a}_t)}{\lambda}\right)\|\pi_\phi(\mathbf{s}_t) - \mathbf{a}_t\|_2^2\right]. \tag{38}$$

Algorithm 2 Offline Transition Collection for IQL

Require: Training scenarios, maximum iteration number $T_{\max}$, and exploratory action strategy
Ensure: Offline dataset $\mathcal{D}$

Initialize $\mathcal{D} \leftarrow \emptyset$
for each training scenario **do**
  for each training episode **do**
    Initialize an ACO scheduler
    Run one ACO iteration and extract the initial state $\mathbf{s}_0$
    for $t = 0$ to $T_{\max} - 2$ **do**
      Extract the current state $\mathbf{s}_t$
      Sample an exploratory action $\mathbf{a}_t$
      Update α, β, and ρ using $\mathbf{a}_t$
      Run the next ACO iteration with the updated parameters
      Compute the reward R_t
      Extract the next state $\mathbf{s}_{t+1}$
      Determine the terminal indicator d_t
      Store $(\mathbf{s}_t, \mathbf{a}_t, R_t, \mathbf{s}_{t+1}, d_t)$ in $\mathcal{D}$
    end for
  end for
end for
return $\mathcal{D}$

3.5. Overall IQACO Procedure

Algorithm 3 summarizes IQACO. In the offline stage, transitions are collected and used to train the IQL networks; the policy π_ϕ is exported. In the online stage, ants construct schedules using current α and β, the best schedule and pheromone matrix are updated with ρ, and π_ϕ outputs parameter adjustments $[\Delta\alpha, \Delta\beta, \Delta\rho]^T$ for the next iteration.

Since the policy input is only five-dimensional and policy inference is performed once per ACO iteration, the additional online overhead of IQACO is small compared with schedule construction and feasibility checking.

4. Simulation Experiments

All algorithms are implemented in C++20 (compiled with `-std=c++20 -O2` using MinGW-w64 GCC 13.2.0) and executed on a workstation with an Intel Core Ultra 9 285H and 64 GB RAM. The IQL module is implemented in Python with PyTorch; the trained policy is exported in ONNX format and loaded via the ONNX Runtime C++ API for CPU inference.

4.1. Scenario Configuration

Fourteen testing scenarios are constructed with 100–240 maritime moving targets and 3–6 satellites, as listed in Table 2. Each original target has 2–4 observation requirements, which are expanded into independent scheduling tasks. The IQL policy is trained on separately generated scenarios with the same parameter ranges but different target distributions and

Algorithm 3 IQL-Bootstrapped Ant Colony Optimization

Require: Training scenarios, test scenario, ACO settings, maximum iteration number $T_{\max}$, and number of ants N_A

Ensure: Best schedule Π^*

Offline training stage
Collect transition dataset $\mathcal{D}$ using Algorithm 2
Train the IQL networks V_ψ, Q_{ϑ_1}, Q_{ϑ_2}, and π_ϕ using $\mathcal{D}$
Export the trained policy π_ϕ

Online scheduling stage
Initialize the ACO scheduler, pheromone matrix $\boldsymbol{\tau}$, parameters α, β, ρ, and best schedule $\Pi^* \leftarrow \emptyset$
for $t = 0$ to $T_{\max} - 1$ **do**
    for $m = 1$ to N_A **do**
        Construct a feasible schedule Π^m using Algorithm 1
    end for
    Evaluate all schedules by the objective function
    Update the best schedule Π^*
    Update the pheromone matrix using ρ
    if $t < T_{\max} - 1$ **then**
        Extract the search state $\mathbf{s}_t$
        Obtain the action $\mathbf{a}_t = \pi_\phi(\mathbf{s}_t)$
        Update α, β, and ρ for the next iteration
    end if
end for
return Π^*

Table 2: Configuration of the experimental scenarios

Scenario	Original targets	Satellites
Scene 01	100	3
Scene 02	120	3
Scene 03	100	4
Scene 04	120	4
Scene 05	140	4
Scene 06	160	4
Scene 07	140	5
Scene 08	160	5
Scene 09	180	5
Scene 10	200	5
Scene 11	180	6
Scene 12	200	6
Scene 13	220	6
Scene 14	240	6

satellite initial conditions, and the 14 scenarios in Table 2 are used only for testing. Targets move within a representative East Asian domain ($\varphi \in [6^\circ, 45^\circ]$, $\lambda \in [105^\circ, 145^\circ]$), with speed

Table 3: IQL training hyperparameters

Parameter	Description	Value
γ	Discount factor	0.99
τ_e	Expectile parameter	0.6
λ	Inverse temperature	5.0
State dimension	Input features	5
Action dimension	Parameter-adjustment action	3
Epochs	Training epochs	200
Learning rate	Adam optimizer	3×10^{-4}

$v_i \sim U(5, 15)$ m/s and direction-persistence probability $\sim U(0.70, 0.95)$ over a 24-h horizon (1-s trajectory step). The satellite cone angle is 25° at 400 km altitude. Key parameters: $d_i = 60$ s, $p_i \in \{1, 2, 3\}$, $E_s^{\max} = 500$ Wh, $P_s^{\text{img}} = 750$ W, $P_s^{\text{att}} = 30$ W, $M_s^{\max} = 2000$ GB, $R_s^{\text{data}} = 4.0$ Gbps. Cloud-affected availability $c_{i,s,k}$ is assigned by latitude: $c = 0.60$ for $|\varphi| < 10°$, $c = 0.70$ for $10° \le |\varphi| < 25°$, $c = 0.80$ for $25° \le |\varphi| < 45°$, and used as cloud-affected availability factors in window screening and objective evaluation.

4.2. Compared Algorithms and Parameter Settings

IQACO is compared with GA, PSO, WOA, and conventional ACO [30, 31, 32]. For fairness, all algorithms use the same scheduling model, objective function, constraint-checking procedure, and evaluation budget; only the search mechanism differs. Each algorithm is executed 20 times per scenario under the same stopping criterion ($N_{\text{FE}} = 20000$). Baseline parameters follow commonly used empirical settings without problem-specific tuning. For IQACO, the initial ACO parameters are $\alpha = 1.0$, $\beta = 2.0$, and $\rho = 0.1$. The IQL policy outputs $[\Delta\alpha, \Delta\beta, \Delta\rho]^T$ with $\Delta\alpha \in [-0.2, 0.2]$, $\Delta\beta \in [-0.4, 0.4]$, and $\Delta\rho \in [-0.1, 0.1]$, clipped to $\alpha \in [1.0, 5.0]$, $\beta \in [1.0, 5.0]$, and $\rho \in [0.1, 0.5]$. IQL hyperparameters are listed in Table 3.

4.3. Results and Discussion

4.3.1. Convergence Analysis

The convergence behavior of the five algorithms is examined on six representative scenarios spanning moderate and large problem scales. Each algorithm is independently executed 20 times per scenario, and the best-so-far objective value is recorded. Figs. 5 and 6 report the mean convergence curves with one-standard-deviation bands, whereas Figs. 7 and 8 show the corresponding final-benefit distributions.

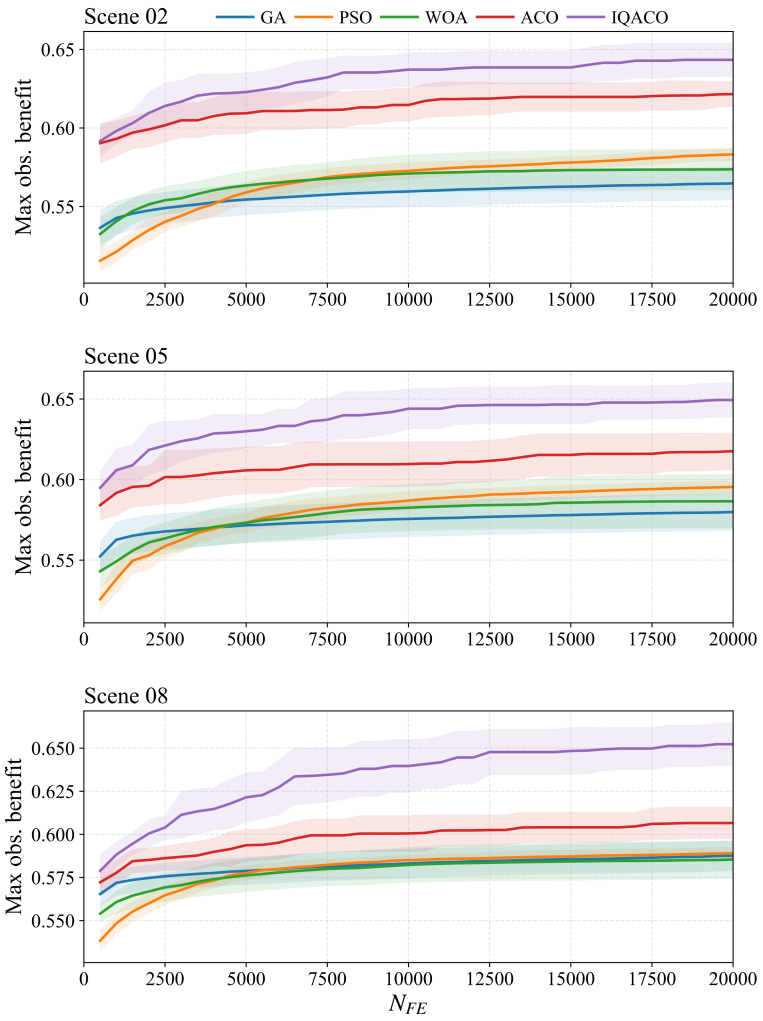

Figure 5: Convergence comparison on Scenes 02, 05, and 08. The solid line denotes the mean best-so-far objective value over 20 independent runs, and the shaded band denotes ± 1 standard deviation.

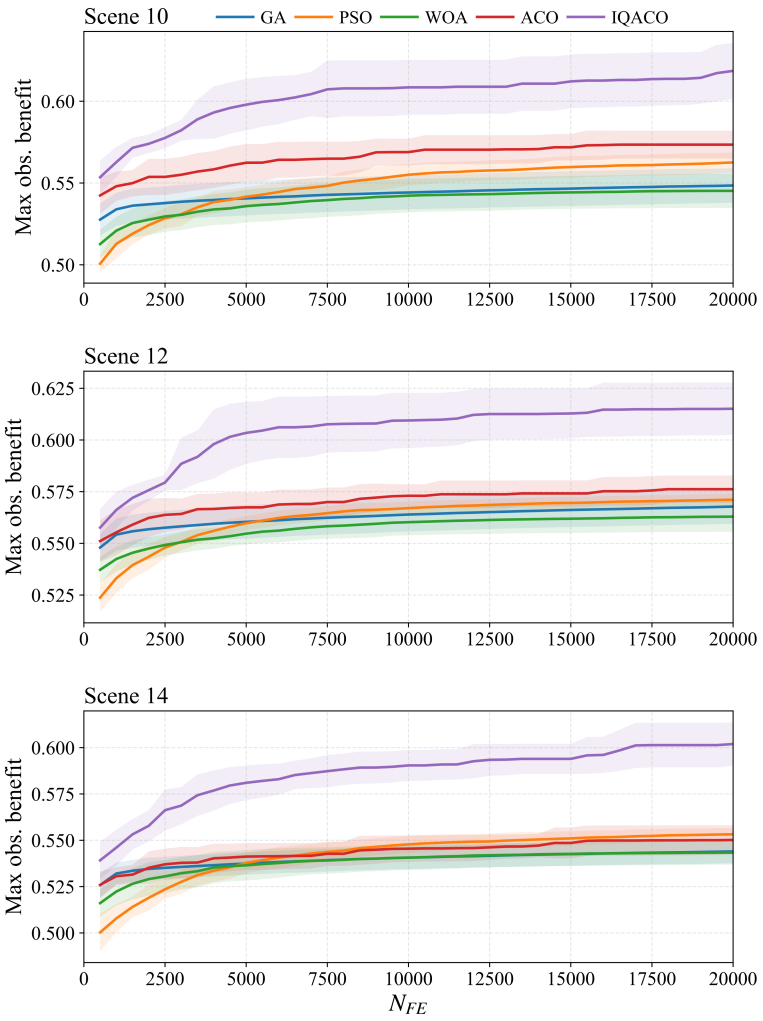

Figure 6: Convergence comparison on Scenes 10, 12, and 14. Solid lines show the mean best-so-far objective value over 20 independent runs, and shaded bands show ± 1 standard deviation.

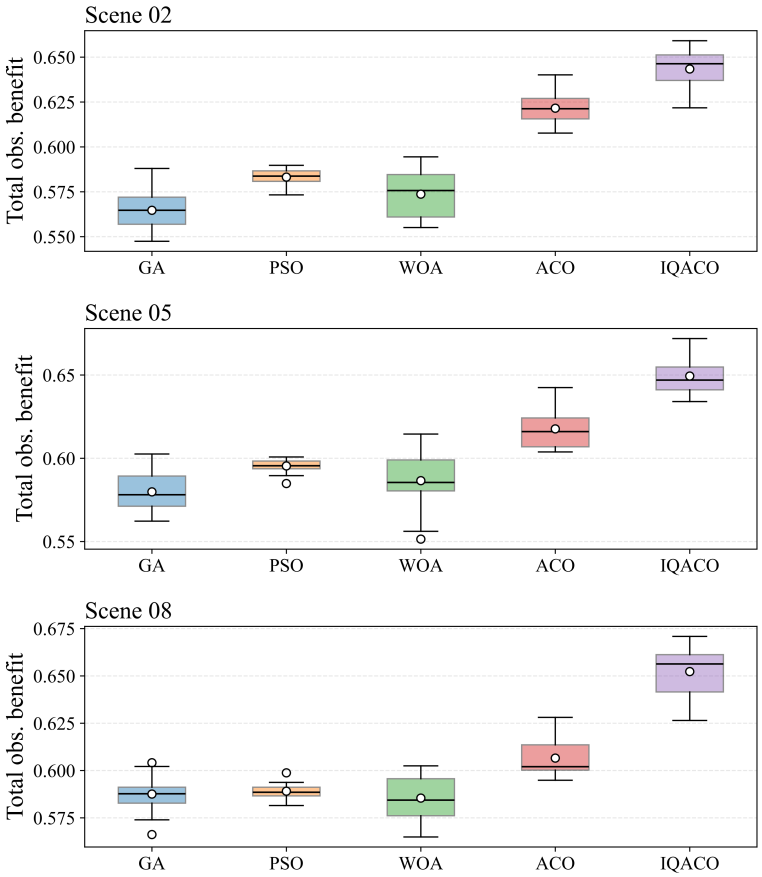

Figure 7: Distribution of the final observation benefit on Scenes 02, 05, and 08. The box denotes the interquartile range, the central mark denotes the median, the triangle denotes the mean, and the whiskers denote the most extreme nonoutlier values.

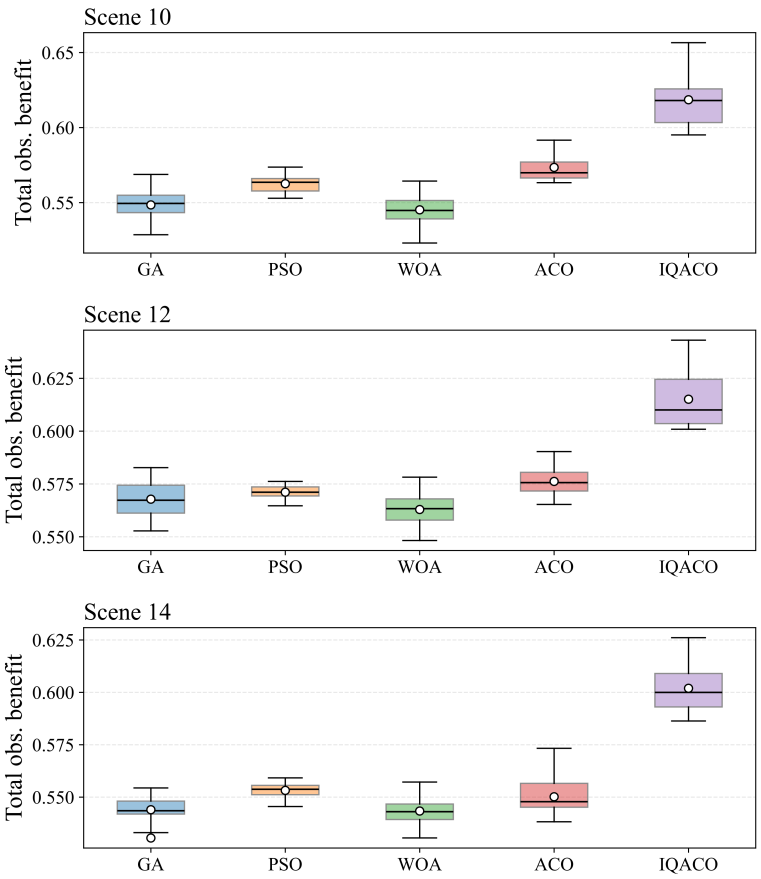

Figure 8: Distribution of the final observation benefit on Scenes 10, 12, and 14. Each box summarizes 20 independent runs; the box denotes the interquartile range, the central mark denotes the median, the triangle denotes the mean, and the whiskers denote the most extreme nonoutlier values.

Figs. 5 and 7 present the convergence behavior and final-benefit distributions for the moderate-scale cases (Scenes 02, 05, and 08). ACO-based methods generally outperform the other competing methods, confirming that constructive search is suitable for this sequence-dependent scheduling problem. Compared with conventional ACO, IQACO reaches higher best-so-far values and shifts the final-benefit distributions upward, indicating that IQL-guided parameter adjustment improves search quality across repeated runs.

Fig. 6 extends the convergence comparison to the larger cases (Scenes 10, 12, and 14). IQACO reaches a higher best-so-far objective level than the four competing algorithms in all three scenes, and the final separation becomes larger as the number of targets and satellites increases. This behavior indicates that fixed ACO parameters become less effective when the feasible-node set and sequence-dependent transitions grow more complex, whereas the IQL controller can adjust the exploration–exploitation balance during the search. The corresponding distributions in Fig. 8 support the same conclusion: IQACO is concentrated at higher observation-benefit levels than the baseline algorithms. Together, these results show that adaptive parameter control is particularly beneficial for large, strongly sequence-dependent scheduling instances.

Table 4 reports the complete statistical results on all 14 scenarios. IQACO obtains the highest mean observation benefit in every scenario, and the Wilcoxon signed-rank test con-

Table 4: Statistical Summary of Final Observation Benefit Over 20 Independent Runs

Scene	GA	PSO	WOA	ACO	IQACO	Gain
01	0.5857 ± 0.0128	0.6003 ± 0.0044	0.5867 ± 0.0154	0.6348 ± 0.0100	$\mathbf{0.6564 \pm 0.0167}^{\ddagger}$	3.40%
02	0.5647 ± 0.0109	0.5832 ± 0.0044	0.5737 ± 0.0134	0.6216 ± 0.0084	$\mathbf{0.6434 \pm 0.0110}^{\ddagger}$	3.51%
03	0.6155 ± 0.0122	0.6159 ± 0.0033	0.6109 ± 0.0108	0.6481 ± 0.0105	$\mathbf{0.6666 \pm 0.0149}^{\ddagger}$	2.85%
04	0.5900 ± 0.0110	0.6028 ± 0.0034	0.5884 ± 0.0145	0.6293 ± 0.0127	$\mathbf{0.6518 \pm 0.0152}^{\ddagger}$	3.58%
05	0.5798 ± 0.0120	0.5954 ± 0.0040	0.5866 ± 0.0172	0.6176 ± 0.0119	$\mathbf{0.6494 \pm 0.0111}^{\ddagger}$	5.15%
06	0.5588 ± 0.0104	0.5774 ± 0.0061	0.5565 ± 0.0118	0.5985 ± 0.0121	$\mathbf{0.6343 \pm 0.0130}^{\ddagger}$	5.98%
07	0.6038 ± 0.0085	0.6052 ± 0.0025	0.6026 ± 0.0065	0.6263 ± 0.0093	$\mathbf{0.6719 \pm 0.0109}^{\ddagger}$	7.28%
08	0.5876 ± 0.0091	0.5891 ± 0.0038	0.5854 ± 0.0111	0.6066 ± 0.0096	$\mathbf{0.6523 \pm 0.0130}^{\ddagger}$	7.53%
09	0.5616 ± 0.0094	0.5679 ± 0.0031	0.5551 ± 0.0080	0.5793 ± 0.0085	$\mathbf{0.6254 \pm 0.0147}^{\ddagger}$	7.96%
10	0.5485 ± 0.0108	0.5626 ± 0.0062	0.5452 ± 0.0107	0.5735 ± 0.0087	$\mathbf{0.6186 \pm 0.0175}^{\ddagger}$	7.86%
11	0.5731 ± 0.0084	0.5717 ± 0.0033	0.5698 ± 0.0067	0.5872 ± 0.0098	$\mathbf{0.6381 \pm 0.0132}^{\ddagger}$	8.67%
12	0.5678 ± 0.0084	0.5711 ± 0.0030	0.5629 ± 0.0074	0.5762 ± 0.0068	$\mathbf{0.6151 \pm 0.0129}^{\ddagger}$	6.75%
13	0.5580 ± 0.0094	0.5654 ± 0.0037	0.5570 ± 0.0080	0.5697 ± 0.0081	$\mathbf{0.6158 \pm 0.0163}^{\ddagger}$	8.09%
14	0.5440 ± 0.0065	0.5532 ± 0.0035	0.5434 ± 0.0067	0.5502 ± 0.0082	$\mathbf{0.6019 \pm 0.0120}^{\ddagger}$	9.40%

‡ indicates that IQACO is significantly better than all baseline algorithms at $p < 0.05$ by the Wilcoxon signed-rank test.

firms significance at $p < 0.05$. The gain over conventional ACO increases from 3.40% in Scene 01 to 9.40% in Scene 14, suggesting that adaptive parameter control becomes more beneficial as the scheduling scale and sequence-dependency complexity increase.

4.3.2. Weight Sensitivity Analysis

Weight sensitivity is evaluated on Scene 01, 04, 07, and 11, which cover different constellation sizes from three to six satellites. Five weight configurations W1–W5 are tested: $(\eta_1, \eta_2, \eta_3) = (0.90, 0.05, 0.05)$, $(0.80, 0.10, 0.10)$, $(0.70, 0.15, 0.15)$, $(0.60, 0.20, 0.20)$, and $(0.50, 0.25, 0.25)$. Fig. 9 presents the mean and standard deviation over 10 runs. IQACO achieves the highest or near-highest values under most configurations. Under benefit-dominated settings, the differences among algorithms are relatively small. As the weights shift toward more balanced multi-objective preferences, the advantage of IQACO becomes more pronounced. This result indicates that adaptive parameter control helps maintain search performance when the objective emphasis changes from benefit maximization to joint consideration of benefit, energy efficiency, and workload balance.

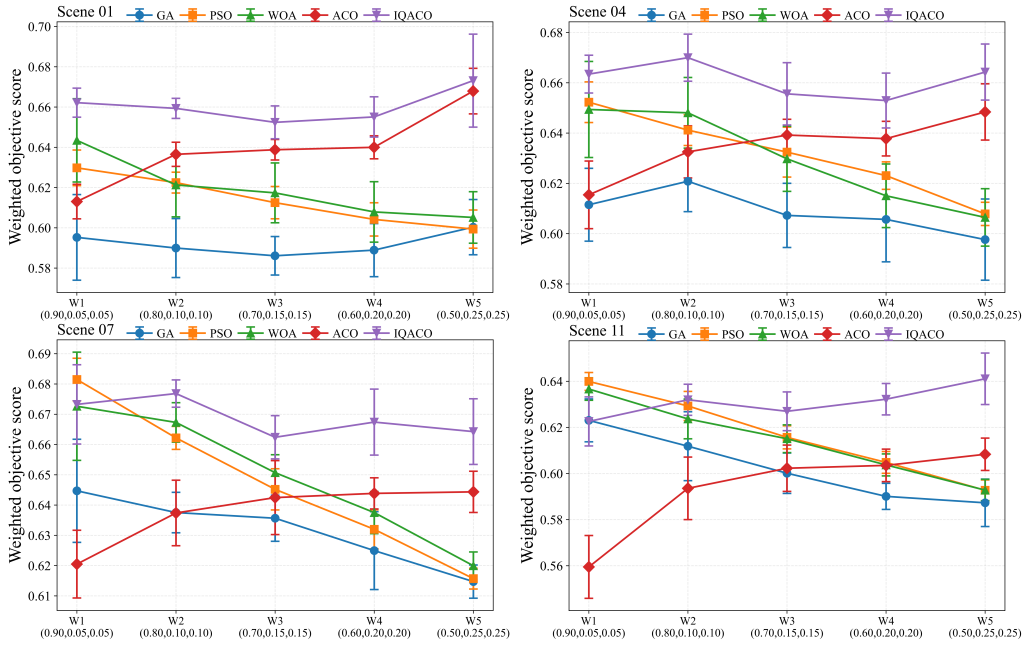

Figure 9: Weight sensitivity results on four representative scenarios under five weight configurations.

4.3.3. IQL Training Convergence Analysis

To further examine the training behavior of the IQL module, the main loss functions and learning statistics are recorded during offline training. Figs. 10–12 show the variations of the total loss, Q loss, value loss, policy loss, action mean-square error, mean Q value, mean V value, mean advantage, and mean action weight with respect to the training epoch.

The training curves show that the IQL module remains stable during offline learning. In the early training stage, the total loss and the main component losses decrease rapidly, indicating that the networks learn useful state–action value information from the offline transition samples. As the number of epochs increases, these losses gradually enter a bounded fluctuation range, and no evident divergence is observed. The decreases in policy loss and action mean-square error indicate that the policy network gradually fits parameter-adjustment actions with higher estimated advantages.

The mean Q value and mean V value also become more stable in the later training stage, suggesting that the value estimates converge to relatively consistent levels. The mean advantage and mean action weight remain within reasonable ranges, which indicates that the advantage-weighted regression does not produce excessively large sample weights. Overall, the IQL module can learn a stable parameter-adjustment policy from offline ACO search trajectories and provide adaptive parameter control for online IQACO search on unseen test scenarios.

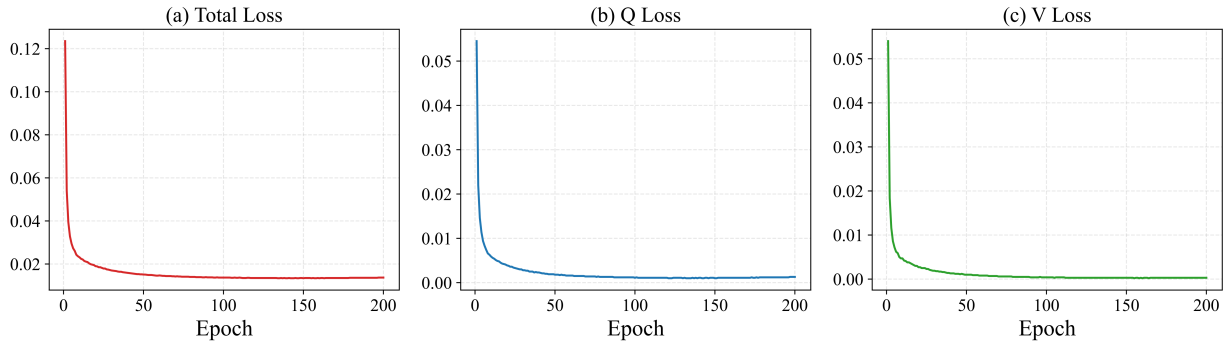

Figure 10: Training loss terms of the IQL module. (a) Total loss. (b) Q loss. (c) V loss.

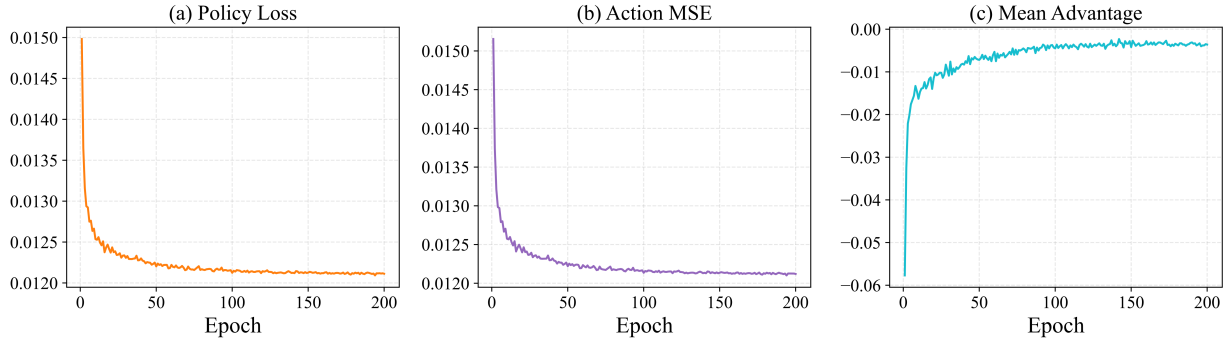

Figure 11: Policy and advantage indicators of the IQL module. (a) Policy loss. (b) Action MSE. (c) Mean advantage.

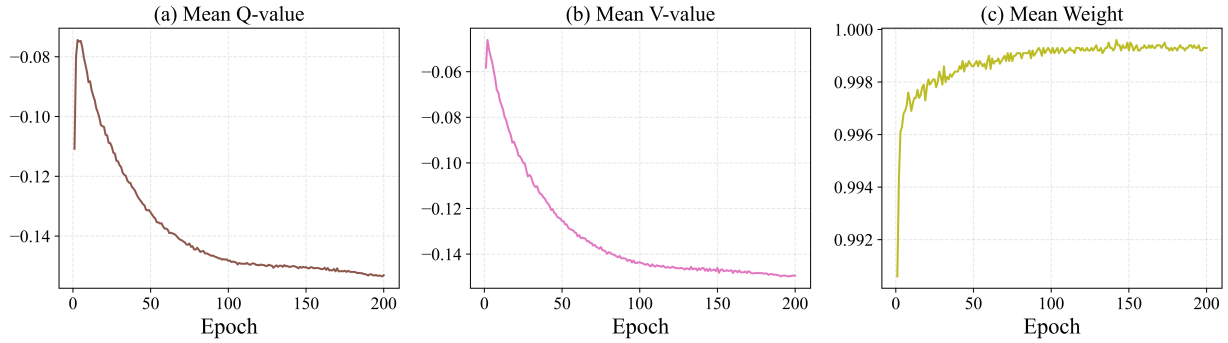

Figure 12: Value estimation and weight statistics. (a) Mean Q-value. (b) Mean V-value. (c) Mean weight.

5. Conclusion

This article addressed multi-satellite maritime moving-target observation scheduling by developing an IQL-guided adaptive ACO framework. The proposed method improves constructive search by learning how to adjust key ACO parameters rather than directly generating scheduling decisions. In this way, the feasibility advantage of ACO-based schedule

construction is preserved, while offline value learning is used to regulate the exploration–exploitation behavior during the search process. Experimental results on 14 scenarios show that IQACO achieves higher objective values and faster convergence than the competing methods. The performance gain becomes more evident in larger-scale scenarios, indicating that adaptive parameter control is particularly useful when the task-window distribution and sequence-dependent constraints become more complex. Weight-sensitivity experiments further show that IQACO maintains competitive performance under different mission-preference settings.

However, the current study is still based on simulated target distributions, simplified cloud-availability modeling, and an offline policy trained within a fixed scenario distribution. Future work will focus on incorporating real AIS trajectories, time-varying cloud fields, and more realistic satellite operation constraints. Transfer learning, online adaptation, and hybrid learning-search mechanisms will also be investigated to improve robustness in operational maritime surveillance applications.

Declaration of competing interest

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.

Data availability

The data and code supporting the findings of this study are available from the corresponding author upon reasonable request.

References

[1] X. Wang, G. Wu, L. Xing, and W. Pedrycz Agile Earth observation satellite scheduling over 20 years: Formulations, methods, and future directions *IEEE Syst. J.*, vol. 15, no. 3, pp. 3881–3892, Sep. 2021.

[2] G. Peng, G. Song, Y. He, B. Deng, and S. Zhao An exact algorithm for agile Earth observation satellite scheduling with time-dependent profits *Comput. Oper. Res.*, vol. 120, Aug. 2020.

[3] G. Peng, G. Song, Y. He, J. Yu, S. Xiang, L. Xing, and P. Vansteenwegen Solving the agile Earth observation satellite scheduling problem with time-dependent transition times *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 52, no. 3, pp. 1614–1625, Mar. 2022.

[4] V. Antuori, N. Beldiceanu, E. Hebrard, and D. Wojtowicz Solving the agile Earth observation satellite scheduling problem In *Proc. 31st Int. Conf. Principles Practice Constraint Program.*, 2025.

[5] A. Chatterjee and R. Tharmarasa Reward factor-based multiple agile satellites scheduling with energy and memory constraints *IEEE Trans. Aerosp. Electron. Syst.*, vol. 58, no. 4, pp. 3090–3103, Aug. 2022.

[6] L. He, B. Liang, J. Li, and M. Sheng Joint observation and transmission scheduling in agile satellite networks *IEEE Trans. Mobile Comput.*, vol. 21, no. 12, pp. 4381–4396, Dec. 2022.

[7] A. M. Mercado-Martínez, B. Soret, and A. Jurado-Navas An energy-efficient learning solution for the Agile Earth Observation Satellite Scheduling Problem In *Proc. ICMLCN*, Barcelona, Spain, 2025, pp. 1–7.

[8] X. Wang, G. Song, R. Leus, and C. Han Robust Earth observation satellite scheduling with uncertainty of cloud coverage *IEEE Trans. Aerosp. Electron. Syst.*, vol. 56, no. 3, pp. 2450–2461, Jun. 2020.

[9] X. Wang, Y. Gu, G. Wu, and J. R. Woodward Robust scheduling for multiple agile Earth observation satellites under cloud coverage uncertainty *Comput. Ind. Eng.*, vol. 156, Jun. 2021.

[10] Y. Chen, J. Xue, W. Gu, and M. Shao An effective Genetic Programming Hyper-Heuristic for Uncertain Agile Satellite Scheduling In *Proc. BigDIA*, Nha Trang, Vietnam, 2025, pp. 311–318.

[11] B. Ferrari, J.-F. Cordeau, M. Delorme, M. Iori, and R. Orosei Satellite Scheduling Problems: A survey of applications in Earth and outer space observation *Comput. Oper. Res.*, vol. 173, 2025.

[12] L. He, X. Liu, G. Laporte, Y. Chen, and Y. Chen An improved adaptive large neighborhood search algorithm for multiple agile satellites scheduling *Comput. Oper. Res.*, vol. 100, pp. 12–25, Dec. 2018.

[13] G. Peng, J. Wang, G. Song, A. Gunawan, L. Xing, and P. Vansteenwegen Branch-and-cut-and-price for agile earth observation satellite scheduling *Eur. J. Oper. Res.*, vol. 326, no. 3, pp. 427–438, 2025.

[14] X. Liu, G. Laporte, Y. Chen, and R. He An adaptive large neighborhood search metaheuristic for agile satellite scheduling with time-dependent transition time *Comput. Oper. Res.*, vol. 86, pp. 41–53, Oct. 2017.

[15] Y. Gu, C. Han, Y. Chen, and W. W. Xing Mission Replanning for Multiple Agile Earth Observation Satellites Based on Cloud Coverage Forecasting *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.*, vol. 15, pp. 594–608, 2022.

[16] H. Wang, W. Huang, S. Magnússon, T. Lindgren, R. Wang, and Y. Song A Strategy Fusion-Based Multiobjective Optimization Approach for Agile Earth Observation Satellite Scheduling Problem *IEEE Trans. Geosci. Remote Sens.*, vol. 62, 2024.

[17] F. Yao, Y. Chen, L. Wang, Z. Chang, P.-Q. Huang, and Y. Wang A bilevel evolutionary algorithm for large-scale multiobjective task scheduling in multiagile Earth observation satellite systems *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 54, no. 6, pp. 3512–3524, Jun. 2024.

[18] B. Wang, Y. Feng, G. Zhang, L. Zhang, and Y. Yang Memetic multiobjective discrete Jaya algorithm for cooperative scheduling of multiple agile Earth observation satellites *IEEE Trans. Aerosp. Electron. Syst.*, vol. 60, no. 6, pp. 8086–8099, Dec. 2024.

[19] Y. Du, T. Wang, B. Xin, L. Wang, Y. Chen, and L. Xing A data-driven parallel scheduling approach for multiple agile Earth observation satellites *IEEE Trans. Evol. Comput.*, vol. 24, no. 4, pp. 679–693, Aug. 2020.

[20] X. Zhou, H. Ma, J. Gu, H. Chen, and W. Deng Parameter adaptation-based ant colony optimization with dynamic hybrid mechanism *Eng. Appl. Artif. Intell.*, vol. 114, Sep. 2022.

[21] A. Herrmann and H. Schaub Reinforcement Learning for the Agile Earth-Observing Satellite Scheduling Problem *IEEE Trans. Aerosp. Electron. Syst.*, vol. 59, no. 5, pp. 5235–5247, Oct. 2023.

[22] J. Chun, W. Yang, X. Liu, G. Wu, L. He, and L. Xing Deep reinforcement learning for the agile Earth observation satellite scheduling problem *Mathematics*, vol. 11, no. 19, Sep. 2023.

[23] A. Jacquet, G. Infantes, N. Meuleau, E. Benazera, S. Roussel, V. Baudoui, and J. Guerra Earth observation satellite scheduling with graph neural networks *arXiv:2408.15041*, 2024.

[24] Z. Liu, W. Xiong, C. Han, and X. Yu Deep reinforcement learning with local attention for single agile optical satellite scheduling problem *Sensors*, vol. 24, no. 19, Oct. 2024.

[25] X. He, J. Xiang, M. Yan, C. Zhang, Z. Xie, and X. Liang Agile Earth observation satellite constellation mission planning based on multi-agent transformer *IEICE Trans. Fundam. Electron., Commun. Comput. Sci.*, vol. E108-A, no. 9, pp. 1316–1319, Sep. 2025.

[26] L. Xu, S. Liu, and S. Qiu An autonomous mission planning method for Earth observation satellites based on reinforcement learning In *Proc. IAF Earth Observation Symp., 76th Int. Astronautical Congr.*, Sydney, Australia, 2025, pp. 605–611.

[27] R. Figueiredo Prudencio, M. R. O. A. Maximo, and E. L. Colombini A Survey on Offline Reinforcement Learning: Taxonomy, Review, and Open Problems *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 8, pp. 10237–10257, Aug. 2024.

[28] A. Riahi Samani, X. Zhao, and F. Chen Distribution shift, generalization and OOD challenge in offline reinforcement learning: A comprehensive survey *Neural Comput. Appl.*, vol. 38, 2026.

[29] I. Kostrikov, A. Nair, and S. Levine Offline reinforcement learning with implicit Q-learning In *Proc. Int. Conf. Learn. Representations*, 2022.

[30] A. Nait Chabane and O. Guenounou An enhanced genetic algorithm for optimized task allocation and planning in heterogeneous multi-robot systems *Complex Intell. Syst.*, vol. 11, 2025.

[31] L. Abualigah Particle Swarm Optimization: Advances, Applications, and Experimental Insights *Comput. Mater. Contin.*, vol. 82, no. 2, pp. 1539–1592, 2025.

[32] L. Han, H. Zhou, Y. Zhang, Y. Wu, and M. Xu An enhanced whale optimization algorithm for task scheduling in edge computing environments *Front. Big Data*, vol. 7, 2024.