drones

MDPI

Article

Risk-Aware Cooperative Planning for Multiple UAVs in Non-Stationary Maritime Missions via a Scenario-Switching-Aware LinUCB Hyper-Heuristic

Jian Wu [1], Shengchang Liu [2,*], Wenxi Ni [3], Junqi Wang [2] and Daming Zhou [2]

[1] Rocket Force University of Engineering, Xi'an 710025, China
[2] School of Astronautics, Northwestern Polytechnical University, Xi'an 710072, China; daming.zhou@nwpu.edu.cn (D.Z.)
[3] Xi'an Institute of Precision Machinery, Xi'an 710077, China
* Correspondence: ericlsc@mail.nwpu.edu.cn; Tel.: +86-18981757277

Highlights

What are the main findings?

- A hierarchical cooperative planning framework is developed for maritime multi-UAV missions by integrating risk-aware path evaluation, rolling task scheduling, and online heuristic selection.
- A switching-aware LinUCB hyper-heuristic is introduced to select among interpretable dispatching rules under non-stationary mission profiles.

What are the implications of the main findings?

- The method improves effective reward robustness and reduces value regret across random maritime mission scenarios.
- The proposed framework keeps the scheduling process interpretable, rather than replacing the full planner with a black box end-to-end model.

Abstract

Maritime unmanned aerial vehicle (UAV) missions such as ship inspection, search and rescue, environmental monitoring, and emergency response often involve multi-wave task releases, time-sensitive deadlines, constrained support vessel positions, and spatially heterogeneous risk. These factors couple task allocation with path planning and make fixed dispatching rules fragile under changing mission profiles. This study develops a hierarchical cooperative planning framework for multiple UAVs over a maritime risk field. A risk-cost A* layer generates feasible routes from support vessels to task points and estimates path length, risk exposure, and sortie duration. A rolling scheduler constructs feasible UAV task candidates, while a scenario-switching-aware LinUCB hyper-heuristic selects online among deadline-first, distance-first, risk-aware, and endurance-balancing rules. A forgetting-update, one-step look-ahead, scenario memory, and lightweight switching detection are used to improve adaptation to mission profile changes. Simulations on a 28×40 maritime grid with two support vessels, six UAVs, 40 tasks, and nine release waves show that the proposed framework achieves the highest average effective reward (370.18), the lowest average value regret (0.61), and a best reward ratio of 0.46 over 24 random scenarios. The results should be interpreted as evidence from an idealized simulation benchmark. The main benefit is improved reward robustness under non-stationary and high-risk profiles, rather than uniform gains across all metrics or direct field-deployment validation.

Academic Editor: Gianfranco Parlangeli

Received: 10 May 2026
Revised: 24 June 2026
Accepted: 28 June 2026
Published: 15 July 2026

Keywords: maritime UAV; multi-task cooperative planning; risk-aware path planning; LinUCB; non-stationary scenario

1. Introduction

Maritime unmanned aerial vehicles (UAVs) are increasingly used for vessel inspection, search and rescue, environmental monitoring, and emergency response. Compared with operations involving a single UAV and a single task, cooperative multi-UAV missions require feasible routes from support platforms to mission points, while also considering task-release sequences, resource occupancy, deadline pressure, and return-to-platform constraints. In maritime environments, task points are often scattered, support vessels are limited, launch and recovery depend on vessel positions, and local risks may vary with weather, sea state, communication quality, restricted areas, or other adverse conditions. These factors increase problem complexity and make path costs and scheduling preferences interdependent. Mission quality, timeliness, risk exposure, and system reward are therefore affected simultaneously.

From a modeling perspective, maritime multi-UAV planning involves at least three coupled levels. First, feasible routes from support vessels to mission points must be generated quickly under a given risk field, with path length, risk exposure, and flight time estimated. Second, task assignment and execution order must be determined under limited UAV resources, sortie duration constraints, and heterogeneous deadlines. Third, when tasks are released in waves and mission profiles evolve over time, scheduling preferences need to adapt to the current situation. Optimizing any one level in isolation makes it difficult to represent the interactions among reward, timeliness, risk, and workload.

The relevant literature can be grouped into task allocation, path planning, and learning-based decision making. Dynamic task allocation studies often use auction mechanisms, priority ranking, matching, genetic algorithms, or combinatorial optimization to describe resource competition. However, many methods simplify path cost as static distance, flight time, or aggregate travel cost, and give less attention to how continuous exposure to a risk field feeds back into dispatching decisions [1,2]. Path planning studies have improved obstacle avoidance, coverage quality, search efficiency, and path smoothness; recent reviews also identify dynamic environments, adaptive coordination, and multi-constraint integration as open problems in multi-UAV path planning [3,4]. In maritime applications, recent work has investigated aerial swarm search and surveillance, maritime search and rescue, multi-region inspection, USV–UAV energy replenishment, ship–drone cooperative search, cooperative vehicle–drone routing, and shipborne drone delivery [5–15]. Drone–port and platform location studies for coastal monitoring further show that platform layout affects persistent monitoring and mission response efficiency [16]. Overall, existing studies provide useful support for maritime applications, route generation, and cooperative scheduling, but many methods remain tied to fixed optimization procedures, specific mission profiles, or static scenario assumptions.

Data-driven and online learning methods offer another way to handle non-stationary mission environments. In maritime UAV systems, deep reinforcement learning has also been used for cooperative data collection and adaptive task area planning in maritime IoT scenarios [17]. Reinforcement-learning-based hyper-heuristics and multi-armed-bandit-based hyper-heuristics can dynamically select search operators or heuristic rules according to feedback [18–20]. LinUCB and related linear contextual bandits score each candidate action by combining its state features with an uncertainty term, forming a lightweight online selection mechanism. In this work, the planner uses those scores to switch among

scheduling rules based on the observed mission state, avoiding the need to train an end-to-end policy [21,22]. For time-varying reward mechanisms, non-stationary bandit studies have introduced change detection, discounting, restart, and anomaly monitoring mechanisms [23–26]. Prior work supports an architecture where the online selection layer does not replace the underlying optimizer but functions as a lightweight rule selector: route search and candidate scheduling still produce feasible solutions, while the bandit layer only chooses the heuristic that guides the current rolling decision. Following this idea, this study does not attempt to learn the entire multi-UAV scheduling process with a monolithic end-to-end model. It retains interpretable base heuristics and uses a scenario-switching-aware LinUCB mechanism to switch adaptively among scheduling preferences, thereby avoiding the offline training burden of a full end-to-end learner while preserving rule interpretability.

Accordingly, this study develops a hierarchical cooperative planning framework with a path layer, a scheduling layer, and a selection layer. The path layer generates feasible routes and estimates key costs. The scheduling layer performs task allocation based on candidate costs. The selection layer observes the current scenario state and selects online among deadline-first, distance-first, risk-aware, and endurance-balancing base heuristics. A forgetting update gradually discounts outdated reward information, so earlier mission profiles do not dominate later decisions. It works alongside one-step look-ahead, scenario memory, and lightweight switching detection to help the planner adapt to changing task profiles.

Figure 1 illustrates the representative maritime multi-UAV mission scenario considered in this study. UAVs depart from support vessel platforms and visit task points over a maritime risk field, while curved lines indicate representative risk-aware routes.

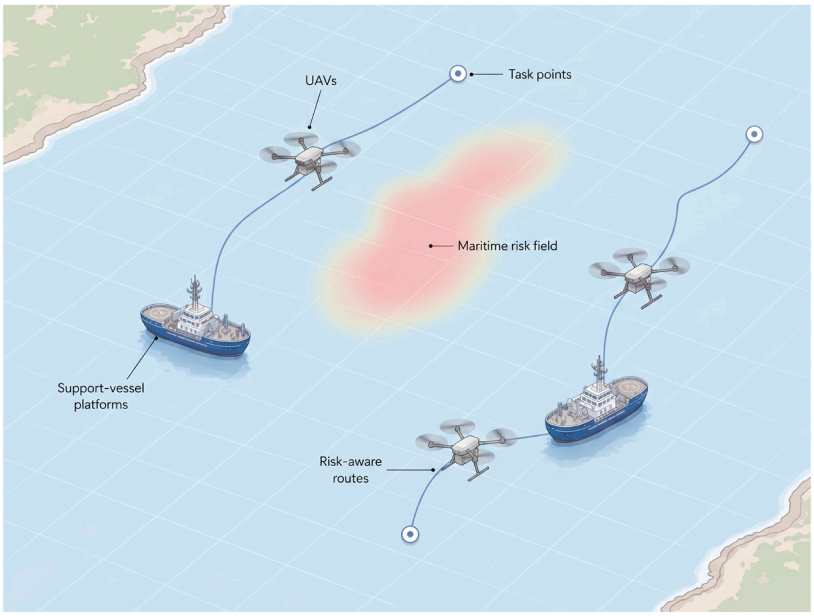

Figure 1. Representative maritime multi-UAV mission scenario.

This scenario motivates the joint consideration of task distribution, path risk, and online scheduling preference in the following method.

The main contributions of this study are summarized as follows:

1. A hierarchical cooperative planning framework is developed for multi-UAV maritime missions in non-stationary environments. The framework jointly considers multi-wave task releases, risk-weighted path costs, task deadlines, and sortie duration constraints within a unified modeling and evaluation setting.

2. A switching-aware LinUCB-based hyper-heuristic scheduling method is designed to adaptively select among four interpretable baseline heuristics. The method integrates forgetting update, one-step look-ahead, scenario memory, and lightweight switching detection to improve adaptation to variations in mission profiles.

3. The proposed framework is evaluated using representative scenario visualization, comparisons over 24 randomly generated scenarios, risk-intensity sensitivity analysis, and mechanism ablation experiments. The results indicate that the proposed method mainly improves the robustness of effective mission value, rather than achieving uniform superiority across all performance metrics.

2. Materials and Methods

2.1. Maritime Risk Field and Task Model

A two-dimensional discrete maritime grid environment is considered and denoted by

$$\mathcal{G} = \{(r, c) \mid 1 \le r \le R_S, 1 \le c \le C_S\}, \tag{1}$$

where R_S and C_S denote the number of grid rows and columns, respectively. Each grid cell has a normalized risk value $\rho(r, c) \in [0, 1]$, which describes the local exposure cost, operational hazard, or environmental risk intensity. The set of support vessel platforms is denoted by $\mathcal{M} = \{1, 2, \ldots, M_0\}$, and their positions are $\mathcal{P}^{\text{base}} = \{p_m^{\text{base}} \mid m \in \mathcal{M}\}$. The UAV set is $\mathcal{U} = \{1, 2, \ldots, U_0\}$, and the task set is $\mathcal{N} = \{1, 2, \ldots, N_0\}$.

Each UAV is assumed to be pre-assigned to a support vessel platform. The platform-affiliation mapping is defined as

$$b : \mathcal{U} \to \mathcal{M}, \tag{2}$$

where $b(u)$ denotes the support vessel to which the UAV u belongs.

Each task $i \in \mathcal{N}$ is described by

$$\mathcal{T}_i = (p_i, r_i, d_i, \tau_i^{\text{srv}}, v_i, q_i, \rho_i^{\text{loc}}, \omega_i), \tag{3}$$

where p_i is the task position, r_i is the release time, namely, the time at which the task i becomes known and eligible for assignment, d_i is the deadline, τ_i^{srv} is the service time, v_i is the task value, q_i is the priority, ρ_i^{loc} is the local risk at the task point, and ω_i is the mission profile type. The local risk at the task point is defined as

$$\rho_i^{\text{loc}} = \rho(p_i). \tag{4}$$

If p_i lies between grid cells, the corresponding local risk value can be obtained by interpolation from neighboring grid risks.

Tasks are not released simultaneously; they appear gradually in multiple waves. By changing the profile structure across release waves, the task stream is designed to exhibit different phases, such as urgent-task-dominated, short-distance-task-dominated, high-risk-task-dominated, and endurance-limited phases.

For each UAV $u \in \mathcal{U}$, let v_u denote its cruise speed and E_u denote its maximum single-sortie duration. UAV u departs from the support vessel position $p_{b(u)}^{\text{base}}$, performs a task, and returns to the same support vessel. If a candidate assignment leads to a total sortie duration larger than E_u, the candidate is removed as infeasible.

2.2. Risk-Aware Path Cost

For a candidate assignment (u, i), the lower-level path planner generates an outbound route $\pi^{\text{out}}_{b(u),i}$ from the platform position $p^{\text{base}}_{b(u)}$ to the task position p_i over the risk field. A* search [27] is used with the following edge cost

$$c(e) = \Delta(e)\big(1 + \lambda_\rho \overline{\rho}(e)\big), \tag{5}$$

where e is an edge between two adjacent grid nodes, $\Delta(e)$ is its geometric length, $\overline{\rho}(e)$ is the average risk value of the two endpoints, and λ_ρ is the risk-weight coefficient. The accumulated path cost is

$$D^{\text{path}}_{u,i} = \sum_{e \in \pi^{\text{out}}_{b(u),i}} c(e), \tag{6}$$

where $D^{\text{path}}_{u,i}$ is used for route selection in the lower-level A* search.

The average risk exposure is defined as

$$R_{u,i} = \frac{1}{\left|\pi^{\text{out}}_{b(u),i}\right|} \sum_{(r,c) \in \pi^{\text{out}}_{b(u),i}} \rho(r, c). \tag{7}$$

The return segment is approximated by the symmetric outbound route. Let $L_{u,i}$ represent the geometric path length from the support vessel $b(u)$ to the task point p_i. Therefore, the total sortie duration of the candidate task is

$$\tau^{\text{sortie}}_{u,i} = \frac{2L_{u,i}}{v_u} + \tau^{\text{srv}}_i. \tag{8}$$

If $\tau^{\text{sortie}}_{u,i} > E_u$, candidate (u, i) is removed before scheduling.

2.3. Rolling Scheduling Timeline and Candidate Utility

Let t^{avail}_u denote the earliest time at which UAV u can be dispatched again. The actual launch time for candidate (u, i) is

$$t^{\text{launch}}_{u,i} = \max(r_i, t^{\text{avail}}_u). \tag{9}$$

The task finish time and recovery time are

$$\begin{cases} t^{\text{finish}}_{u,i} = t^{\text{launch}}_{u,i} + \frac{L_{u,i}}{v_u} + \tau^{\text{srv}}_i \\ t^{\text{rec}}_{u,i} = t^{\text{finish}}_{u,i} + \frac{L_{u,i}}{v_u} \end{cases}. \tag{10}$$

The tardiness is

$$\Gamma_{u,i} = \max\left(0, t^{\text{finish}}_{u,i} - d_i\right), \tag{11}$$

and the on-time indicator is

$$z_{u,i} = \begin{cases} 1, & \Gamma_{u,i} = 0 \\ 0, & \Gamma_{u,i} > 0 \end{cases}. \tag{12}$$

At decision time t, the task-processing indicator is

$$\Theta_i = \begin{cases} 1, & \text{task } i \text{ has been processed} \\ 0, & \text{task } i \text{ has not been processed} \end{cases}. \tag{13}$$

The active task set is

$$\mathcal{N}_a(t) = \{i \in \mathcal{N} \mid r_i \le t, \Theta_i = 0\}. \tag{14}$$

Let $x_{u,i}(t) \in \{0,1\}$ indicate whether task i is assigned to UAV u at the current decision step. Rolling scheduling must satisfy

$$\begin{cases} \sum\limits_{u \in \mathcal{U}} x_{u,i}(t) \le 1, \ \forall i \in \mathcal{N}_a(t) \\ t_{u,i}^{\text{launch}} \ge r_i, \ \forall (u,i) \\ \tau_{u,i}^{\text{sortie}} \le E_u, \ \forall (u,i) \end{cases}. \tag{15}$$

To describe the local tradeoff among task value, risk, and timeliness, the candidate utility is defined as

$$J_{u,i} = \eta_{u,i} v_i - \lambda_1 R_{u,i} - \lambda_2 \tau_{u,i}^{\text{sortie}} - \lambda_3 \Gamma_{u,i}, \tag{16}$$

where λ_1, λ_2, and λ_3 are penalty weights for exposure risk, sortie duration, and tardiness, respectively. The tardiness discount coefficient $\eta_{u,i}$ is

$$\eta_{u,i} = \begin{cases} 1, & \Gamma_{u,i} = 0 \\ \eta_0, & \Gamma_{u,i} > 0 \end{cases}, \tag{17}$$

where $\eta_0 \in [0,1]$ is the reward discount for delayed task completion. The utility $J_{u,i}$ is used for local candidate ranking. Subsequent base-heuristic sorting and online hyper-heuristic selection provide a rolling approximation to this local reward structure.

3. Proposed Switching-Aware LinUCB Hyper-Heuristic Method

3.1. Hierarchical Cooperative Planning Framework

The proposed framework is organized into three layers: a path layer, a scheduling layer, and an online selection layer. For each candidate UAV–task pair, the path layer performs risk-aware path evaluation from a support vessel platform to a task point and estimates the corresponding path length, risk exposure, sortie duration, and finish time. These path cost indicators are then passed to the scheduling layer, where infeasible candidates are screened out, and feasible task assignments are ranked under multiple base heuristics. Based on the active task set, UAV states, and scenario context, the online selection layer selects the most suitable heuristic for the current decision step and updates its rule statistics using execution feedback. This architecture preserves the coupling between task allocation and path cost while avoiding a monolithic black box learner for the entire dispatching process. As shown in Figure 2, the proposed hierarchical cooperative planning method comprises several key components.

3.2. Base Heuristic Operators

To preserve interpretability, four base heuristic operators are defined as $\mathcal{H} = \{h_1, h_2, h_3, h_4\}$. They represent: (1) a deadline-first operator, which prioritizes tasks with tighter deadlines; (2) a distance-first operator, which prioritizes tasks with lower path length and sortie cost; (3) a risk-aware operator, which prefers candidates with lower exposure risk; and (4) an endurance-balancing operator, which mitigates workload imbalance among UAVs.

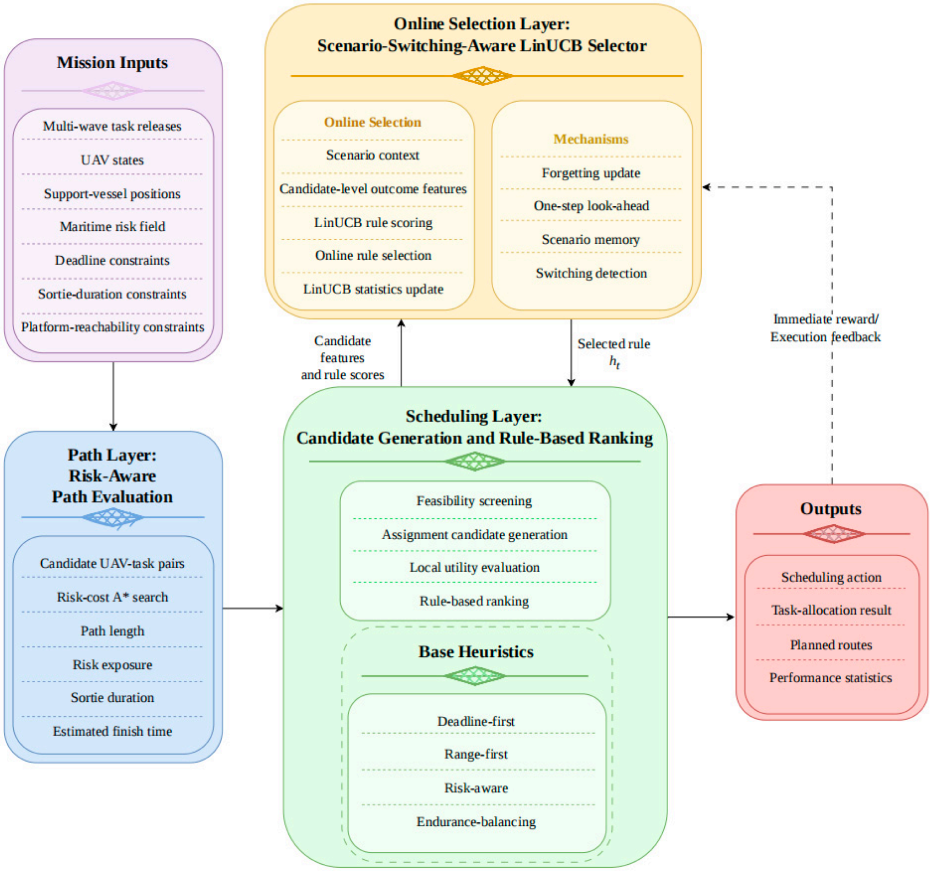

Figure 2. Overall framework of the proposed hierarchical cooperative planning method.

To connect these rules with the upper hyper-heuristic layer, the candidate scores of the four base operators are written as

$$\begin{cases} g_{u,i}^{(1)} = -\left(d_i - t_{u,i}^{\text{finish}}\right) \\ g_{u,i}^{(2)} = -\widetilde{L}_{u,i} - \widetilde{\tau}_{u,i}^{\text{sortie}} \\ g_{u,i}^{(3)} = -R_{u,i} \\ g_{u,i}^{(4)} = -\kappa_1 \widetilde{\tau}_{u,i}^{\text{sortie}} - \kappa_2 \widetilde{F}_u - \kappa_3 \widetilde{B}_u \end{cases}, \tag{18}$$

where variables with tildes are normalized within the current candidate set to remove scale differences among path length, sortie duration, and load indicators. Parameters κ_1 κ_2, and κ_3 are the weights of sortie duration, accumulated load, and availability busyness in the endurance-balancing operator. The first score represents task urgency by negative slack, the second compresses path length and sortie cost, the third suppresses high-risk candidates, and the fourth mitigates local resource imbalance through accumulated flight load F_u and busyness B_u.

At decision step t, let $\mathcal{C}_t$ be the active candidate set after feasibility filtering. Each base operator ranks this candidate set and selects the candidate with the highest score under its own rule, which is therefore defined as

$$(u_h^*, i_h^*) = \arg\max_{(u,i)\in\mathcal{C}_t} g_{u,i}^{(h)}. \tag{19}$$

The four base heuristics serve as both comparison baselines and the action space of the hyper-heuristic layer. The online learning layer does not output a complete schedule directly; it selects the rule that is more suitable for the current scenario.

3.3. Scenario-Switching-Aware LinUCB Hyper-Heuristic

LinUCB is used as the basic online decision architecture [21,22]. At the scheduling step t, the system first constructs a global state vector

$$\mathbf{x}_t = [\chi_t^{\text{pending}}, \chi_t^{\text{released}}, \chi_t^{\text{urgency}}, \chi_t^{\text{risk}}, \chi_t^{\text{energy}}, \chi_t^{\text{load}}]^\top. \tag{20}$$

The six components represent the pending task ratio, released task ratio, urgent task ratio, average local risk, average sortie intensity, and load dispersion, respectively. The average local risk is computed from ρ_i^{loc} in the active task set, which is defined as

$$\chi_t^{\text{risk}} = \frac{1}{|\mathcal{N}_a(t)|} \sum_{i \in \mathcal{N}_a(t)} \rho_i^{\text{loc}}. \tag{21}$$

Let $\mathbf{e}(\omega_i)$ denote the one-hot encoding of mission profile type ω_i. The current mission profile proportion is

$$\mathbf{s}_t^{\text{prof}} = \frac{1}{|\mathcal{N}_a(t)|} \sum_{i \in \mathcal{N}_a(t)} \mathbf{e}(\omega_i). \tag{22}$$

For each base heuristic h, the global state, the outcome features of the first candidate proposed by the rule, and the mission profile information are concatenated into a joint state feature, which is written as

$$\boldsymbol{\Phi}_{t,h} = \left[1, \mathbf{x}_t^\top, z_{u_h^\star, i_h^\star}, \widetilde{v}_{i_h^\star}, \widetilde{q}_{i_h^\star}, \rho_{i_h^\star}^{\text{loc}}, R_{u_h^\star, i_h^\star}, \widetilde{\tau}_{u_h^\star, i_h^\star}^{\text{sortie}}, \widetilde{L}_{u_h^\star, i_h^\star}, \mathbf{e}(\omega_{i_h^\star})\right]^\top. \tag{23}$$

Here $\widetilde{q}_{i_h^\star}$ is the normalized task priority, and $z_{u_h^\star, i_h^\star}$ indicates whether the candidate is estimated to have completed on time. The joint state feature can also be extended with resource busyness, wave index, and mission profile type according to the experimental setting. The wave index denotes the ordinal label of the task release batch to which a task belongs. It is used to indicate the current stage of the mission stream and to help the selector distinguish early, middle, and later task arrival patterns.

For each base heuristic h, the LinUCB score is

$$\psi_{t,h} = \hat{\boldsymbol{\theta}}_h^\top \boldsymbol{\Phi}_{t,h} + \alpha_t \sqrt{\boldsymbol{\Phi}_{t,h}^\top \mathbf{A}_h^{-1} \boldsymbol{\Phi}_{t,h}} + \beta_r \hat{y}_{t,h}^{\text{imm}} + \beta_l \hat{y}_{t,h}^{\text{look}} + b_{t,h}^{\text{reg}}, \tag{24}$$

where $\hat{\boldsymbol{\theta}}_h = \mathbf{A}_h^{-1} \mathrm{b}_h$ is the current parameter estimate, $\alpha_t \sqrt{\boldsymbol{\Phi}_{t,h}^\top \mathbf{A}_h^{-1} \boldsymbol{\Phi}_{t,h}}$ is the upper-confidence exploration term, $\hat{y}_{t,h}^{\text{imm}}$ is a one-step immediate reward estimate, $\hat{y}_{t,h}^{\text{look}}$ is a one-step look-ahead reward estimate, and $b_{t,h}^{\text{reg}}$ is the profile consistency bias. Parameters α_t, β_r, and β_l denote the exploration intensity, the immediate reward fusion weight, and the look-ahead fusion weight, respectively.

The immediate reward estimate uses the analytical utility of the candidate ranked first by the corresponding operator and is computed as

$$\hat{y}_{t,h}^{\text{imm}} = J_{u_h^\star, i_h^\star}. \tag{25}$$

The one-step look-ahead term does not solve a complete dynamic program. After executing the current candidate $(u_h^\star, i_h^\star)$, it conducts a lightweight approximation over the feasible candidate set $\mathcal{C}_{t+1}^{(h)}$ of the next scheduling step

$$\hat{y}_{t,h}^{\text{look}} = \max_{(u',j) \in \mathcal{C}_{t+1}^{(h)}} \left(\eta_{u',j} v_j - \lambda_1 R_{u',j} - \lambda_2 \tau_{u',j}^{\text{sortie}} - \lambda_3 \Gamma_{u',j}\right). \tag{26}$$

This one-step correction evaluates whether the current action will substantially reduce the feasible candidate space in the next step.

3.4. Lightweight Enhancement Mechanisms

To handle non-stationary maritime mission profiles, the selection layer incorporates four lightweight enhancement mechanisms.

First, forgetting updates decay old statistics with the forgetting coefficient $\gamma \in (0, 1]$ to avoid being dominated by early-stage experience, which is described as

$$\begin{cases} \mathbf{A}_h \leftarrow \gamma \mathbf{A}_h + \boldsymbol{\phi}_{t,h} \boldsymbol{\phi}_{t,h}^{\top} \\ \mathbf{b}_h \leftarrow \gamma \mathbf{b}_h + r_t \boldsymbol{\phi}_{t,h} \end{cases}. \tag{27}$$

Second, one-step look-ahead considers not only the immediate reward of the current step, but also the impact of the current action on the next-step resource occupation and candidate structure, thereby reducing myopic decisions.

Third, scenario memory stores learning statistics for different mission profile states and attempts to restore or mix historical memory when similar profiles reappear. Let the historical scenario memory bank be

$$\mathcal{B}_{\text{mem}} = \left\{ \mathbf{s}_{\text{prof}}^{(k)}, \mathbf{A}_h^{(k)}, \mathbf{b}_h^{(k)} \right\}_{k=1}^{K_{\text{mem}}}, \tag{28}$$

where $\mathbf{s}_{\text{prof}}^{(k)}$ is the profile proportion vector of the kth historical scenario, and K_{mem} is the memory bank size. The similarity weight between the current profile vector $\mathbf{s}_t^{(k)}$ and a historical profile is

$$\varpi_{t,k} = \frac{\exp\left(-\| \mathbf{s}_t^{\text{prof}} - \mathbf{s}_{\text{prof}}^{(k)} \|_1 / \sigma_{\text{mem}} \right)}{\sum_{j=1}^{K_{\text{mem}}} \exp\left(-\| \mathbf{s}_t^{\text{prof}} - \mathbf{s}_{\text{prof}}^{(j)} \|_1 / \sigma_{\text{mem}} \right)}, \tag{29}$$

where $\sigma_{\text{mem}} > 0$ is the similarity scale parameter and $\sum_{k=1}^{K} \varpi_{t,k} = 1$. Memory recovery uses a weighted mixing form, which is computed as

$$\begin{cases} \mathbf{A}_h \leftarrow (1 - \xi_t) \mathbf{A}_h + \xi_t \sum_{k=1}^{K_{\text{mem}}} \varpi_{t,k} \mathbf{A}_h^{(k)} \\ \mathbf{b}_h \leftarrow (1 - \xi_t) \mathbf{b}_h + \xi_t \sum_{k=1}^{K_{\text{mem}}} \varpi_{t,k} \mathbf{b}_h^{(k)} \end{cases}, \tag{30}$$

where $\xi_t \in [0, 1]$ is the memory-injection strength. If the current profile differs greatly from historical profiles, ξ_t is reduced to avoid erroneous transfer.

Fourth, lightweight switching awareness constructs a change-point score from global-state change, mission profile distribution change, wave transition, and reward-prediction error. The global-state change and profile change are

$$\begin{cases} \Delta_t^{\text{state}} = \frac{1}{d_x} \| \mathbf{x}_t - \mathbf{x}_{t-1} \|_1 \\ \Delta_t^{\text{prof}} = \frac{1}{d_{\text{prof}}} \| \mathbf{s}_t^{\text{prof}} - \mathbf{s}_{t-1}^{\text{prof}} \|_1 \end{cases}, \tag{31}$$

where d_x and d_{prof} denote the dimensions of the global state and profile vector, respectively. If w_i is the wave index of task i, the dominant wave index in the active task set is

$$w_t^{\text{dom}} = \underset{w \in \mathcal{W}}{\text{argmax}} \sum_{i \in \mathcal{N}_a(t)} \mathbf{1}(w_i = w), \tag{32}$$

where $\mathcal{W}$ is the set of all wave indices, $1(\cdot)$ is the indicator function. The dominant-wave transition indicator is

$$\Delta_t^{\text{wave}} = \mathbf{1}\left(w_t^{\text{dom}} \neq w_{t-1}^{\text{dom}}\right). \tag{33}$$

Let the immediate reward obtained at step t be y_t, and let the selected rule's immediate reward estimate before execution be $\hat{y}_{t,h_t}^{\text{imm}}$. The reward deviation is

$$\varepsilon_t^y = \left| y_t - \hat{y}_{t,h_t}^{\text{imm}} \right|. \tag{34}$$

The exponentially smoothed deviation is

$$\bar{\varepsilon}_t^y = (1 - \lambda_{\text{sur}})\bar{\varepsilon}_{t-1}^y + \lambda_{\text{sur}}\varepsilon_t^y, \tag{35}$$

where $\lambda_{\text{sur}} \in (0, 1]$ is the smoothing coefficient. The reward prediction error term is

$$\Delta_t^{\text{sur}} = \min(1, \left|\bar{\varepsilon}_t^y\right|). \tag{36}$$

The change-point score is

$$\mathcal{S}_t = \zeta_{\text{state}}\Delta_t^{\text{state}} + \zeta_{\text{prof}}\Delta_t^{\text{prof}} + \zeta_{\text{wave}}\Delta_t^{\text{wave}} + \zeta_{\text{sur}}\Delta_t^{\text{sur}}, \tag{37}$$

where ζ_{state}, ζ_{prof}, ζ_{wave}, and ζ_{sur} are fusion weights. If $\mathcal{S}_t$ exceeds the switching threshold $\delta_{\mathcal{S}}$, soft reset and short-term exploration amplification are performed. With soft-reset ratio $\mu_t \in [0, 1]$, the statistics are corrected as

$$\begin{cases} \mathbf{A}_h \leftarrow (1 - \mu_t)\mathbf{A}_h + \mu_t\lambda_0\mathbf{I} \\ \mathbf{b}_h \leftarrow (1 - \mu_t)\mathbf{b}_h + \mu_t\gamma_0\boldsymbol{\theta}_h^{\text{prior}} \end{cases}, \tag{38}$$

where λ_0 is the regularization scale, γ_0 is the prior pullback coefficient, and $\boldsymbol{\theta}_h^{\text{prior}}$ is the estimated reward coefficient vector for heuristic operator h, learned from previous feedback. If $S_t > \delta_s$, the exploration scale is adjusted for the next W_s decision steps as

$$\alpha_{t+\tau} = \alpha_0(1 + \kappa_s),\ \tau = 0, 1, \ldots, W_s - 1, \tag{39}$$

where α_0 is the base exploration coefficient and $\kappa_s > 0$ is the exploration amplification factor after switching. This design makes the selector increase exploration after a significant change in scenario statistics.

The immediate reward after selecting an action at the current step is defined as

$$y_t = c_z z_t + c_v\widetilde{v}_t + c_s\widetilde{s}_t - c_R R_t - c_\tau\widetilde{\tau}_t^{\text{sortie}} - c_L\widetilde{L}_t - c_\Gamma\widetilde{\Gamma}_t, \tag{40}$$

where z_t is the on-time indicator, $\widetilde{v}_t$ is the normalized task reward, $\widetilde{s}_t$ is the positive-slack reward term, R_t is the exposure risk, and $\widetilde{\tau}_t^{\text{sortie}}$, $\widetilde{L}_t$, and $\widetilde{\Gamma}_t$ are normalized sortie duration, path length, and tardiness, respectively. Parameters c_z, c_v, c_s, c_R, c_τ, c_L, and c_Γ are reward function weights.

3.5. Algorithm Execution Procedure

Figure 3 shows the execution process of the scenario-switching-aware LinUCB hyper-heuristic planning procedure. It links the rolling closed loop to the corresponding equations in candidate generation, risk-aware path search, rule-based ranking, online selection, and execution update. Algorithm 1 gives the corresponding pseudocode.

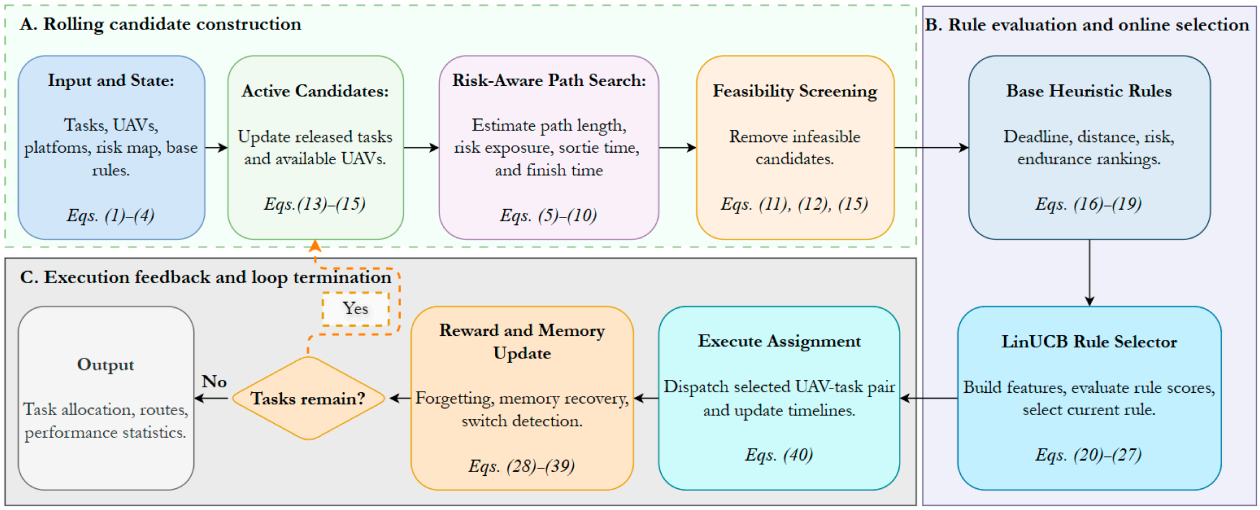

Figure 3. Execution process of the scenario-switching-aware LinUCB hyper-heuristic planning procedure.

Algorithm 1 Scenario-Switching-Aware LinUCB Hyper-Heuristic Planning Procedure

Input: Task set $\mathcal{N}$, UAV set $\mathcal{U}$, support-vessel platform set $\mathcal{M}$, maritime grid $\mathcal{G}$, risk map ρ, and base heuristic rule set $\mathcal{H}$

Output: Task-allocation sequence, planned routes, and scheduling performance statistics

Initialize: LinUCB statistics $\{\mathbf{A}_h, \mathbf{b}_h\}_{h\in\mathcal{H}}$, scenario memory buffer, UAV states, and task states
while unprocessed tasks remain **do**
  if no released and unassigned task is active at the current decision time **then**
    Advance the decision time to the next task-release time
    continue
  end if
  Construct the active candidate set $\mathcal{C}_\sqcup$, according to the current time and resource states
  for all $(u, i) \in \mathcal{C}_t$ **do**
    Call the risk-aware path-search module to compute $L_{u,i}$, $R_{u,i}$, $\tau_{u,i}^{\text{sortie}}$, and $t_{u,i}^{\text{finish}}$
  end for
  Remove infeasible candidates that violate sortie-duration, deadline, or platform-reachability constraints
  if $\mathcal{C}_t = \varnothing$ **then**
    Advance the decision time according to the release and resource-availability schedule
    continue
  end if
  for all $h \in \mathcal{H}$ **do**
    Rank $\mathcal{C}_t$, using rule h, and obtain the top-ranked candidate $(u_h^\star, i_h^\star)$ and the local score $g_{u,i}^{(h)}$
    Construct the global scenario context $\mathbf{x}_t$, and the rule-specific joint feature vector $\boldsymbol{\Phi}_{t,h}$
    Compute the rule score $\psi_{t,h}$
  end for
  Select the base heuristic with the highest score:

$$h_t = \text{argmax}_{h\in\mathcal{H}} \psi_{t,h}.$$

  Execute the task assignment corresponding to rule h_t, and update the UAV timelines and task set
  Compute the immediate reward y_t, and update the LinUCB statistics of the selected rule
  Apply forgetting update, scenario-memory recovery, and lightweight switching-aware adjustment
end while
Return: task-allocation results, planned routes, and scheduling performance statistics

At each scheduling step, an active candidate set is constructed from released tasks, available UAVs, and platform constraints. The risk-aware path search module calculates path length, risk exposure, sortie duration, and finish time for each candidate. Infeasible candidates are removed before the four base heuristics rank the candidate set separately.

The LinUCB selector chooses the current base rule online based on the scenario state, candidate-level outcome features, and rule scores. The task assignment associated with the selected rule is then executed. The UAV timelines, task states, immediate reward, and online statistics are updated. Forgetting-based update, scenario memory recovery, and lightweight switching detection are applied. The loop continues while unprocessed tasks remain. The procedure returns task allocation results, planned routes, and scheduling performance statistics.

3.6. Feasibility and Numerical Stability Analysis

Property 1 (Constraint-feasibility preservation)**.** *If the algorithm selects task assignment actions only from the feasible candidate set $\mathcal{C}_t$ at each decision step, the resulting schedule satisfies unique task assignment, release-time constraints, and UAV sortie duration constraints.*

By construction, any candidate $(u, i) \in \mathcal{C}_t$ must satisfy task release, task unassigned status, UAV availability, and $\tau_{u,i}^{\text{sortie}} \leq E_u$. Algorithm 1 removes candidates that violate sortie duration constraints, deadline constraints, or platform reachability. The LinUCB layer selects only among rankings induced by the remaining feasible candidates. Therefore, an executed action cannot originate from an infeasible candidate. After a task is executed, it is removed from the unprocessed set, so the same task is not assigned repeatedly. The constructive rolling procedure, therefore, preserves these hard constraints.

Property 2 (Positive definiteness and bounded update of online statistics)**.** *If the initialized matrix satisfies $\mathbf{A}_h = \lambda_0 \mathbf{I}$ with $\lambda_0 > 0$, the forgetting coefficient satisfies $\gamma \in (0, 1]$, the soft reset ratio satisfies $\mu_t \in [0, 1]$, and the memory injection strength satisfies $\xi_t \in [0, 1]$, then the statistical matrix $\mathbf{A}_h$ of each base heuristic remains positive semidefinite during updates and remains invertible under regularization.*

Since $\boldsymbol{\phi}_{t,h} \boldsymbol{\phi}_{t,h}^{\top}$ is positive semidefinite, the forgetting update $\gamma \mathbf{A}_h + \boldsymbol{\phi}_{t,h} \boldsymbol{\phi}_{t,h}^{\top}$ does not destroy positive semi-definiteness. The soft reset term $(1 - \mu_t)\mathbf{A}_h + \mu_t \lambda_0 \mathbf{I}$ is a convex combination of a positive semidefinite matrix and a positive definite matrix. The scenario memory term is also a weighted mixture of historical statistical matrices. As long as the regularization scale $\lambda_0 > 0$ is retained, the matrix used to compute $\mathbf{A}_h^{-1}$ does not degenerate to the zero matrix. This property gives the confidence term in the LinUCB score a clear numerical meaning.

If the joint feature vector $\boldsymbol{\phi}_{t,h}$ and the immediate reward y_t are normalized or clipped, the parameter estimate $\hat{\boldsymbol{\theta}}_h = \mathbf{A}_h^{-1} \mathbf{b}_h$ will not be unboundedly amplified by a single abnormal candidate. Feature scale control is therefore a necessary condition for numerical stability in online updates.

3.7. Complexity and Scalability Analysis

The main computational cost comes from candidate generation and risk-aware path evaluation. Let N denote the number of tasks, U the number of UAVs, H the number of base heuristics, and V and E the number of grid nodes and edges. At scheduling step t, let K_t be the number of feasible UAV-task candidates and D the feature dimension used by the online selector. Risk-aware path evaluation for one candidate has the same order as an A*-type graph search and can be written as $O(|E| \log |V|)$ with a priority queue

implementation. If all candidate paths are recomputed, the path evaluation cost at step t is bounded by $O(K_t|E|\log|V|)$. The heuristic ranking layer adds $O(HK_t \log K_t)$. The LinUCB scoring and update cost is $O(HD^2)$ when inverse statistics are updated incrementally, and can rise to $O(HD^3)$ if matrix inversions are recomputed. Under the present scale, repeated path evaluation is the main bottleneck. Larger maritime regions or denser task sets would benefit from path caching, incremental graph updates, and parallel candidate evaluation.

4. Results and Analysis

4.1. Simulation Setup

The simulations use an idealized 28 × 40 maritime grid with two support vessel platforms, six UAVs, and 40 tasks. The tasks are released in nine waves, and the baseline risk-intensity coefficient is set to 1.0. In this setting, the average task value is approximately 12.38, the average deadline is approximately 2.96 h, the average release time is approximately 1.31 h, and the average local risk is approximately 0.51. This setting produces time pressure, risk heterogeneity, and resource competition simultaneously.

The proposed scenario-switching-aware LinUCB hyper-heuristic is compared with five baselines: deadline-first, distance-first, risk-aware, endurance-balancing, and auction-regret. The four fixed heuristics represent different engineering preferences. The auction-regret baseline first computes the candidate UAV utility for each task and then forms a task bid from the opportunity loss between the best and second-best candidates.

To evaluate differences in timeliness, reward quality, risk control, and robustness, repeated experiments are conducted over multiple random scenarios. In total, 24 random scenarios are generated using different random seeds, corresponding to independent instances with different task spatial distributions, release times, and local risk structures. The metrics are reported as averages.

The experiments are designed as controlled, idealized simulations. This setting provides a reproducible testbed for examining how the scheduling mechanism behaves under multi-wave task releases and changing mission profiles. In the main comparison, random scenarios are generated by varying task locations, release times, and local risk distributions under the same scale parameters. The experiments therefore evaluate comparative robustness under controlled non-stationarity, rather than full operational performance in real maritime environments. Additional models for sea state, platform motion, communication quality, launch and recovery procedures, and regulatory constraints are required before field deployment.

To evaluate the performance of the proposed method, all simulation experiments were conducted in MATLAB R2020b on a computer equipped with an AMD Ryzen 7 5800H CPU with Radeon Graphics and 16 GB RAM.

4.2. Evaluation Metrics

Let $\mathcal{N}_{\text{sch}}$ be the set of tasks that are eventually assigned and completed. Let $\mathcal{N}_{\text{ot}}$ be the set of tasks completed on time, and let $\mathcal{N}_{\text{late}}$ be the set of delayed but still scheduled tasks. They satisfy

$$\begin{cases} \mathcal{N}_{\text{sch}} = \mathcal{N}_{\text{ot}} \cup \mathcal{N}_{\text{late}} \\ \mathcal{N}_{\text{ot}} \cap \mathcal{N}_{\text{late}} = \varnothing \end{cases}. \tag{41}$$

If task i is finally executed by UAV $u(i)$, the on-time indicator is

$$z_i = \begin{cases} 1, & i \in \mathcal{N}_{\text{ot}} \\ 0, & i \notin \mathcal{N}_{\text{ot}} \end{cases}. \tag{42}$$

The path risk of task i is

$$R_i^{\text{sch}} = R_{u(i),i}. \tag{43}$$

Unscheduled tasks are regarded as not completed on time and have zero effective reward contribution. The main evaluation metrics are as follows.

The average on-time completion rate is

$$r_{ot} = \frac{1}{N_0} \sum_{i=1}^{N_0} z_i. \tag{44}$$

The effective task reward is

$$V_{\text{eff}} = \sum_{i \in \mathcal{N}_{\text{ot}}} v_i + \eta_0 \sum_{i \in \mathcal{N}_{\text{late}}} v_i. \tag{45}$$

The average exposure risk of the schedule is

$$\overline{R}_{\text{sch}} = \frac{1}{|\mathcal{N}_{\text{sch}}|} \sum_{i \in \mathcal{N}_{\text{sch}}} R_i^{\text{sch}}. \tag{46}$$

For average value regret, let $\mathcal{A}_{\text{comp}}$ be the set of compared methods and N_s be the number of random scenarios. In scenario s, the best observed effective reward among all compared methods is

$$V_{\max}^{(s)} = \max_{a \in \mathcal{A}_{\text{comp}}} V_{\text{eff}}^{(s,a)}. \tag{47}$$

The value regret of method a in scenario s is

$$n_{\text{reg}}^{(s,a)} = V_{\max}^{(s)} - V_{\text{eff}}^{(s,a)}. \tag{48}$$

The average value regret of method a is

$$\overline{n}_{\text{reg}}^{(a)} = \frac{1}{N_s} \sum_{s=1}^{N_s} n_{\text{reg}}^{(s,a)}. \tag{49}$$

The best reward ratio is

$$r_W^{(a)} = \frac{1}{N_s} \sum_{s=1}^{N_s} \mathbf{1}\left(V_{\text{eff}}^{(s,a)} = V_{\max}^{(s)}\right). \tag{50}$$

Average value regret and best reward ratio jointly describe reward robustness over random scenarios. A smaller average value regret indicates that a method is closer to the best observed reward in different instances. A larger best reward ratio indicates that the method more frequently obtains the highest reward in a random scenario. Since effective reward is computed over a discrete task set, if multiple methods achieve the same best effective reward in one scenario, they are treated as tied for the best methods for that scenario.

4.3. Single-Scenario Planning Results

Figure 4 shows the spatial planning result in a representative single scenario. Task points are mainly distributed around two high-risk hotspots and their surrounding regions. The paths show local detours and route separation. The lower path module does not connect support vessels and task points using only Euclidean distance. Instead, it jointly incorporates path risk and travel cost into the candidate cost calculation.

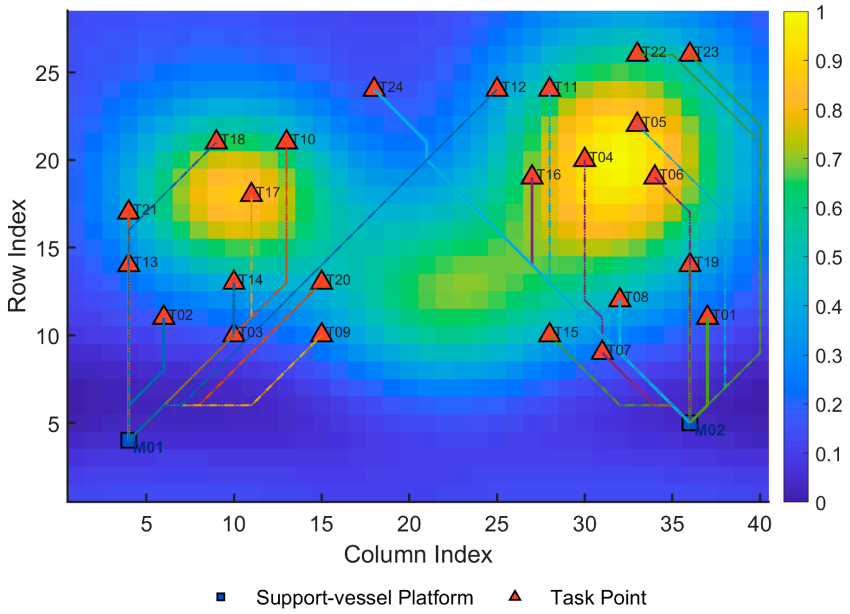

Figure 4. Spatial planning results in a single scenario.

Figure 5 gives the UAV task timeline for the same scenario. Horizontal bars represent task execution intervals, dashed segments represent return or idle movement, and red markers indicate task deadlines. The timeline illustrates resource occupation: tasks assigned to the same UAV must be executed serially, and the feasibility of subsequent tasks depends on previous service time and return time. Some tasks are close to their deadlines, indicating that the case contains clear deadline pressure. This provides the scenario basis for online rule selection.

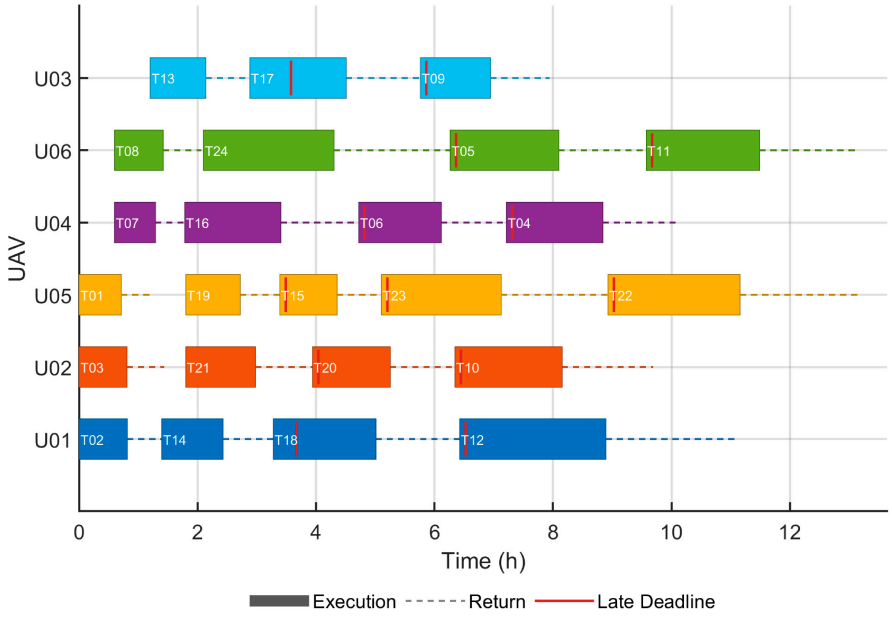

Figure 5. UAV task timeline in the same scenario.

4.4. Multi-Scenario Comparison

Figure 6 and Table 1 present the comparison over 24 random scenarios, with V_{eff}, $\overline{n}_{\text{reg}}$ and time in Table 1 expressed as mean ± standard deviation (SD). The proposed method attains a V_{eff} of 370.18 ± 3.44—the highest among the six tested methods—and the lowest $\overline{n}_{\text{reg}}$ of 0.61 ± 0.85. All main comparisons are accompanied by SDs and 95% confidence-

interval (CI) half-widths. Although the improvement in average effective reward over the distance-first method is modest, the best reward ratio is slightly higher at 0.46. These results indicate that the main advantage of the proposed selector lies not in a large single-instance reward increase, but in a higher probability of approaching the best observed reward across random scenarios.

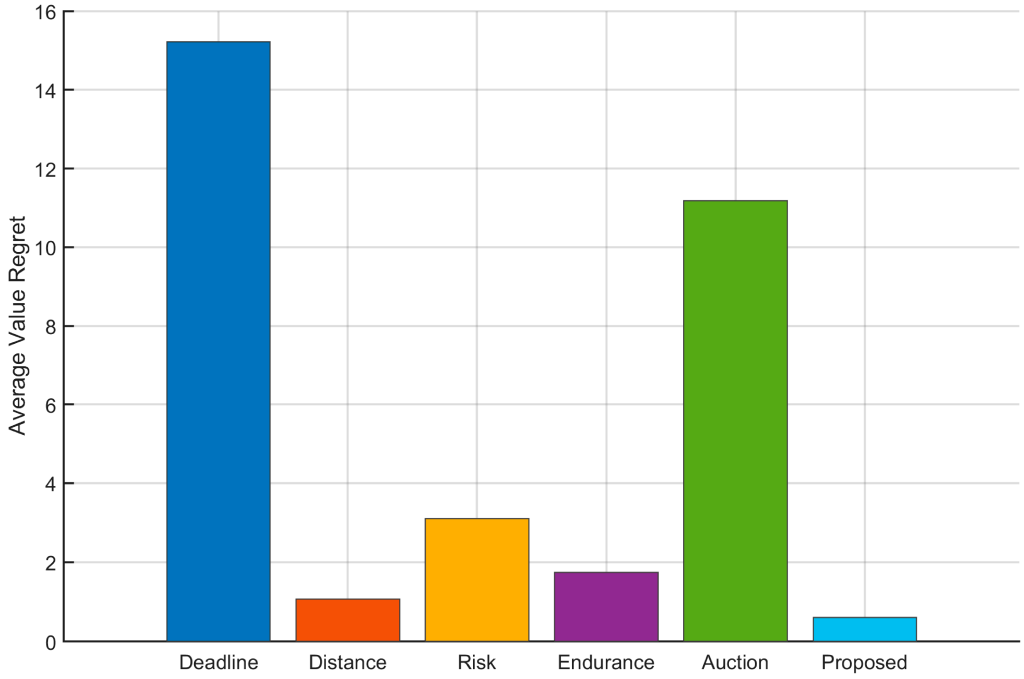

Figure 6. Average value regret comparison over 24 random scenarios.

Table 1. Multi-random-scenario comparison statistics.

Method	r_{ot}	r_W	$\bar{R}_{sch}$	95% CI Half-Width	V_{eff}	$\bar{n}_{reg}$	Time (s)
Deadline-first	0.1500	355.58	0.2262	0.90	355.58 ± 2.24	15.21 ± 1.95	0.069 ± 0.005
Distance-first	0.2969	369.72	0.2298	1.10	369.72 ± 2.74	1.07 ± 1.44	0.066 ± 0.003
Risk-aware	0.2833	367.70	0.2108	1.57	367.70 ± 3.91	3.10 ± 3.04	0.067 ± 0.004
Endurance-balancing	0.2885	369.05	0.2298	1.28	369.05 ± 3.19	1.75 ± 1.81	0.067 ± 0.004
Auction-regret	0.1760	359.62	0.2262	1.02	359.62 ± 2.55	11.18 ± 2.74	0.063 ± 0.006
Proposed method	0.2969	370.18	0.2238	1.38	370.18 ± 3.44	0.61 ± 0.85	0.632 ± 0.028

To examine whether the observed reward differences are robust across random scenarios, as shown in Table 2, the reward advantage is statistically supported against the deadline-first, risk-aware, endurance-balanced, and auction-regret methods. The difference from the distance-first method is small and not statistically significant (mean difference 0.46, $p = 0.241$). The overall finding is therefore interpreted as improved value robustness under non-stationary task profiles, rather than uniform dominance over all methods.

Table 2. Paired comparison between the proposed method and others.

Method	Reward Difference	*p*-Value	Supported Improvement?
Deadline-first	14.60 ± 0.84	<0.001	Yes
Distance-first	0.46 ± 0.77	0.241	No
Risk-aware	2.49 ± 1.19	<0.001	Yes
Endurance-balancing	1.13 ± 0.96	0.021	Yes
Auction-regret	10.56 ± 1.19	<0.001	Yes

4.5. Sensitivity to Risk Intensity

Figure 7 and Table 3 show how the risk-intensity coefficient affects effective task reward. When the risk intensity is 0.7, the proposed method obtains an average effective reward of 355.58, lower than that of the distance-first, endurance-balancing, and risk-aware methods. This result indicates that when risk pressure is weak, fixed heuristics can already obtain good rewards, and the marginal benefit of online selection is limited.

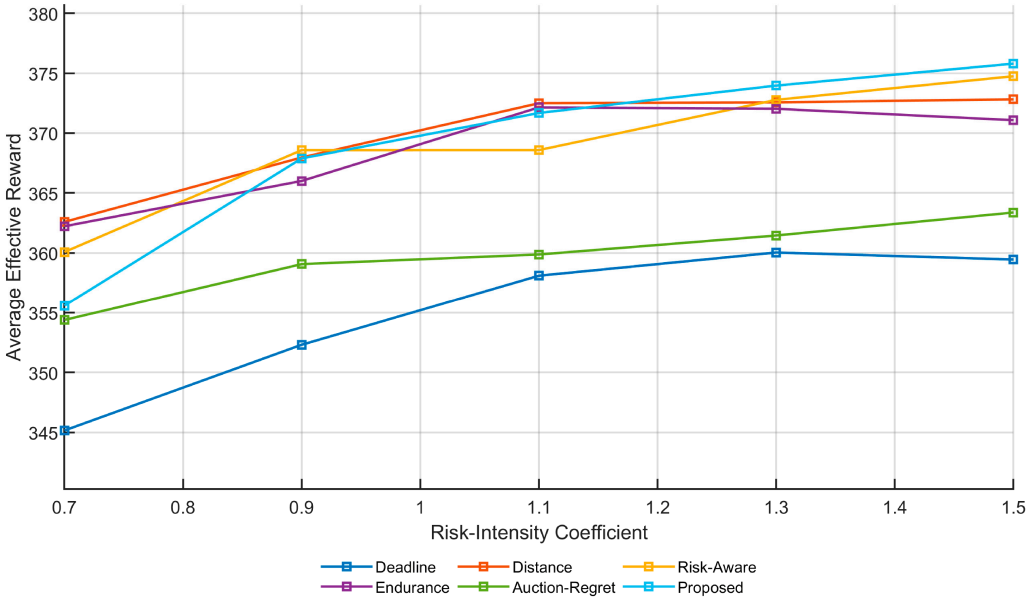

Figure 7. Sensitivity of average effective reward to the risk-intensity coefficient.

Table 3. Average effective reward under different risk-intensity coefficients.

Method	0.7	0.9	1.1	1.3	1.5
Deadline-first	345.15	352.31	358.09	360.03	359.44
Distance-first	362.58	367.96	372.49	372.56	372.81
Risk-aware	360.07	368.57	368.58	372.77	374.74
Endurance-balancing	362.22	366.00	372.14	372.02	371.08
Auction-regret	354.38	359.06	359.86	361.45	363.37
Proposed method	355.58	367.89	371.68	373.96	375.79

As risk intensity increases, the proposed method approaches, and in some cases exceeds, the fixed rules. When the risk intensity is 1.3, the proposed method reaches an average effective reward of 373.96, higher than all other methods. When the risk intensity is 1.5, it obtains 375.79 and remains competitive. This trend indicates that, under stronger risk pressure, a single fixed preference is more likely to be affected by scenario changes, whereas online selection can adaptively adjust among different preferences.

4.6. Mechanism Ablation Analysis

Additional mechanism ablation is conducted in a non-stationary switching scenario. The ablation experiment focuses on a repeated four-profile cycle in which distance-compact, risk-sensitive, deadline-critical, and endurance-limited profiles appear sequentially and repeatedly. This scenario is designed to test how well the online selection layer adapts to scenario changes. In Table 4, "LinUCB with forgetting and look-ahead" denotes a baseline that includes a forgetting update and one-step look-ahead but does not explicitly use lightweight switching awareness or scenario memory correction. "No forgetting" and "No look-ahead" remove the corresponding sub-mechanisms. "Default switching-aware"

uses the default parameters of the switching-aware LinUCB. "Proposed method" uses the parameter-tuned setting adopted in the previous comparisons.

Table 4. Mechanism ablation results under repeated four-profile switching.

Variant	r_{ot}	V_{eff}	$\lvert V_{\text{eff}} - V_{\text{eff}}^{\max} \rvert$	r_W	Time (s)
LinUCB with forgetting and look-ahead	0.1042	394.70	0.41	0.583	0.745
No forgetting	0.1042	394.70	0.41	0.583	0.737
No look-ahead	0.1042	394.82	0.29	0.667	0.735
Default switching-aware	0.1042	395.00	0.11	0.916	0.763
Proposed method	0.1042	395.11	0	1	0.768

Table 4 shows that the on-time completion rates of all variants are almost identical. This suggests that in this high-pressure scenario, on-time completion is mainly determined by task density, release wave interval, and sortie duration constraints, rather than by a single online-learning mechanism. The difference among variants is mainly reflected in the effective reward and the relative gap to the best reward. The default switching-aware variant reduces the average relative gap to 0.11, and the parameter-tuned setting further reaches the best observed value in this group. Switching awareness and parameter tuning improve the average effective reward under repeated mechanism shifts and persistent resource pressure.

5. Discussion

The scenario-switching-aware LinUCB selector primarily enhances reward robustness, not all performance indicators uniformly. Across the tested random scenarios, the proposed framework achieved the highest average effective reward and the lowest average value regret. Online rule selection helped the scheduler approach the best observed reward more frequently when mission profiles changed, but the improvement in absolute reward was moderate, and the on-time completion rate remained close to the strongest fixed baseline. The method is therefore a reward quality and robustness enhancement for non-stationary maritime scheduling, not a universal replacement for fixed heuristic rules.

In maritime multi-UAV planning, task allocation is coupled with path-dependent risk, sortie duration constraints, platform reachability, and heterogeneous deadlines. A fixed rule performs well when the mission profile matches its design preference, but becomes less reliable as task urgency, spatial distribution, risk intensity, and endurance pressure change. The proposed online selector retains interpretable heuristic rules and uses the current scenario state and candidate-level outcome features to switch among them, preserving interpretability while adapting the scheduling preference online.

The risk-intensity and ablation results clarify the applicable conditions. At low risk levels, several fixed heuristics already achieve competitive rewards, so the marginal benefit of online selection is limited. Higher risk intensity strengthens the interaction between route exposure and scheduling preference, making adaptive rule selection more valuable. Ablation results further show that switching awareness, scenario memory, look-ahead correction, and parameter adjustment mainly improved reward-oriented decisions. Their effect on the on-time completion rate was limited, since timeliness was tightly constrained by task release intervals, UAV availability, platform positions, and single-sortie endurance limits.

From a deployment perspective, the framework is better interpreted as a mission-level planning layer than as a low-level flight control or safety-critical autonomy module. The computationally heavier components, including risk-aware path evaluation, rolling candidate construction, and online rule selection, are more suitable for support vessel-

side or edge-assisted execution. UAVs can receive task assignments, route segments, and replanning commands from this higher-level planner. This execution mode reduces onboard computational burden, but it requires communication-aware update intervals, synchronization among UAVs and support vessels, and fallback rules when maritime links are temporarily unavailable.

Real maritime UAV operations also involve weather variability, wind disturbance, GNSS degradation, intermittent communication, launch and recovery constraints, battery limits, and operational regulations. These factors are not solved by the scheduling layer alone. They should be treated as external constraints or additional state layers when the planner is embedded into an operational system. The planner can provide mission-level task and route decisions, while trajectory tracking, collision avoidance, navigation resilience, and safety certification remain the responsibility of dedicated onboard or operational subsystems.

Several limitations should be acknowledged. The experiments are conducted on an idealized maritime grid, and the maritime risk field is an abstract proxy for spatially varying mission difficulty. Depending on the application layer, it may represent adverse sea state, restricted areas, communication degradation, or operational risk. It is not a calibrated oceanographic, meteorological, or communication model. The current experiments also exclude moving support vessel uncertainty, dynamic obstacle avoidance, energy replenishment logistics, and detailed collision-avoidance dynamics. These simplifications make the scheduling mechanism easier to isolate and compare, but they limit deployment-oriented claims. Future extensions should couple the planner with data-driven sea-state layers, communication quality maps, moving-platform models, launch and recovery constraints, and energy replenishment constraints.

6. Conclusions

This study investigated a cooperative planning problem for multiple UAVs in non-stationary maritime mission environments. A hierarchical framework was developed by combining risk-aware path evaluation, rolling task scheduling, and a scenario-switching-aware LinUCB hyper-heuristic selection layer. The framework incorporates multi-wave task releases, path exposure risk, deadline constraints, platform reachability, and single-sortie endurance limits within a unified simulation setting. By retaining interpretable base heuristic rules and selecting among them online, the method preserves the coupling between task allocation and path cost while avoiding a monolithic black box learning formulation of the entire dispatching process.

The simulation results showed stronger reward-oriented robustness under the current experimental setting. Over 24 random scenarios, the framework reported an average effective reward of 370.18, an average value regret of 0.61, and a best reward ratio of 0.460. The advantage was more evident under medium- and high-risk conditions, where fixed scheduling preferences were more easily affected by changes in the mission profile.

The results should be interpreted with caution. The proposed framework does not uniformly improve all indicators. Its on-time completion rate remains close to the strongest fixed baseline, and its runtime exceeds that of fixed greedy rules because of the added online selection and update steps. Furthermore, these conclusions are limited to the idealized simulation benchmark used in this study. Validation aimed at deployment would require more realistic maritime dynamics, communication constraints, launch and recovery procedures, and larger scenario sets. Future work will focus on dynamic risk fields, communication-aware coordination, moving support vessels, explicit launch and recovery constraints, heterogeneous UAV fleets, stronger baselines, and larger-scale validation.

Author Contributions: Conceptualization, J.W. (Jian Wu), S.L., J.W. (Junqi Wang) and D.Z.; methodology, J.W. (Jian Wu), W.N. and D.Z.; software, J.W. (Jian Wu), S.L., W.N. and D.Z.; validation, S.L., J.W. (Junqi Wang) and D.Z.; formal analysis, J.W. (Jian Wu), and J.W. (Junqi Wang); investigation, J.W. (Jian Wu), S.L., W.N. and J.W. (Junqi Wang); resources, J.W. (Jian Wu), S.L., W.N. and J.W. (Junqi Wang); data curation, J.W. (Jian Wu), S.L. and D.Z.; writing—original draft preparation, J.W. (Jian Wu), S.L. and D.Z.; writing—review and editing, J.W. (Jian Wu), S.L., W.N. and D.Z.; visualization, J.W. (Jian Wu), S.L. and D.Z.; supervision, J.W. (Jian Wu), W.N. and D.Z.; project administration, J.W. (Jian Wu), J.W. (Junqi Wang), W.N. and D.Z.; funding acquisition, J.W. (Jian Wu), W.N. and D.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Shaanxi Province Key Research and Development Plan (2024CY-JJQ-59), Xi'an Science and Technology Plan Project (25KGYB00030), Zhoushan Science and Technology Plan Project (2023C03005).

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Acknowledgments: During the preparation of this manuscript, the authors used ChatGPT image generation (GPT-Image-2.0) by OpenAI to create the illustrative maritime multi-UAV mission scenario shown in Figure 1. The authors reviewed and edited the generated output and take full responsibility for the content of this publication.

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

Abbreviations

The following abbreviations are used in this manuscript:

A*	A-star search algorithm
LinUCB	Linear upper confidence bound
UAV	Unmanned aerial vehicle
USV	Unmanned surface vehicle

References

1. Alqefari, S.; Menai, M.E.B. Multi-UAV Task Assignment in Dynamic Environments: Current Trends and Future Directions. *Drones* **2025**, *9*, 75. [CrossRef]
2. Cao, M.; Peng, J.; Yang, L.; Yang, Y. Multi-UAV Task Allocation and Path Planning in Emergency Rescue Scenarios with Uncertain Requirements. *J. Ind. Manag. Optim.* **2025**, *21*, 6107–6133. [CrossRef]
3. Rahman, M.; Sarkar, N.I.; Lutui, R. A Survey on Multi-UAV Path Planning: Classification, Algorithms, Open Research Problems, and Future Directions. *Drones* **2025**, *9*, 263. [CrossRef]
4. Braßel, H.; Zeh, T.; Lindner, M.; Fricke, H. Risk-Aware UAV Trajectory Optimization Using Open Urban GIS Data and Target Level of Safety Constraints. *Drones* **2025**, *9*, 666. [CrossRef]
5. Yang, S.; Lin, D.; He, S.; Hussain, I.; Seneviratne, L. Aerial Swarm Search for GNSS-Denied Maritime Surveillance. *IEEE Trans. Aerosp. Electron. Syst.* **2024**, *60*, 3442–3453. [CrossRef]
6. Li, Y.; Chen, W.; Fu, B.; Wu, Z.; Hao, L. A Hierarchical Decoupling Task Planning Method for Multi-UAV Collaborative Multi-Region Coverage with Task Priority Awareness. *Drones* **2025**, *9*, 575. [CrossRef]
7. Zhang, W.; Chen, G.; Yang, Z. Auction- and Pheromone-Based Multi-UAV Cooperative Search and Rescue in Maritime Environments. *Drones* **2025**, *9*, 794. [CrossRef]
8. Hong, Y.; Wu, Q. Dual-Stage Collaborative Path Planning and Task Allocation for UAV Swarms in Complex Maritime Rescue Environments. *Reliab. Eng. Syst. Saf.* **2026**, *268*, 111974. [CrossRef]
9. Yang, J.; Zhao, L.; Peng, B. Joint Path Planning and Energy Replenishment Optimization for Maritime USV–UAV Collaboration under BeiDou High-Precision Navigation. *Drones* **2025**, *9*, 746. [CrossRef]
10. Xiong, Y.; Tian, H.; Tang, J.; Jin, J.; Shen, X. Task Planning and Optimization for Multi-Region Multi-UAV Cooperative Inspection. *Drones* **2025**, *9*, 762. [CrossRef]
11. Sun, S.; Zhang, H.; Dong, E. Multi-UAV Path Planning Based on IACO and Improved YOLOV8 Perception for Maritime Search and Rescue. *EURASIP J. Wirel. Commun. Netw.* **2025**, *2025*, 94. [CrossRef]

12. Hou, X.; Ji, M.; Kong, L.; Gao, Z.; Wu, D.; Zheng, J. Collaborative Path Optimization of Ship and Multiple Drones for Maritime Search. *Eur. J. Oper. Res.* **2026**, *332*, 693–710. [CrossRef]
13. Li, X.; Zhang, H. Cooperative Path Planning Optimization for Ship-Drone Delivery in Maritime Supply Operations. *Complex Intell. Syst.* **2025**, *11*, 232. [CrossRef]
14. Jiang, J.; Dai, Y.; Yang, F.; Zhang, R.; Ma, Z. The Cooperative Vehicle Routing Problem with Drones. *IEEE Trans. Syst. Man Cybern. Syst.* **2026**, *56*, 292–306. [CrossRef]
15. Morin, M.; Abi-Zeid, I.; Quimper, C.-G. Ant Colony Optimization for Path Planning in Search and Rescue Operations. *Eur. J. Oper. Res.* **2023**, *305*, 53–63. [CrossRef]
16. Sun, J.; Shu, S.; Hu, H.; Deng, Y.; Li, Z.; Zhou, S.; Liu, Y.; Dang, M.; Huang, W.; Hou, Z.; et al. Location Optimization of Unmanned Aerial Vehicle (UAV) Drone Port for Coastal Zone Management: The Case of Guangdong Coastal Zone in China. *Ocean Coast. Manag.* **2025**, *262*, 107576. [CrossRef]
17. Fu, X.; Huang, X.; Pan, Q.; Pace, P.; Aloi, G.; Fortino, G. Cooperative Data Collection for UAV-Assisted Maritime IoT Based on Deep Reinforcement Learning. *IEEE Trans. Veh. Technol.* **2024**, *73*, 10333–10349. [CrossRef]
18. Lagos, F.; Pereira, J. Multi-Armed Bandit-Based Hyper-Heuristics for Combinatorial Optimization Problems. *Eur. J. Oper. Res.* **2024**, *312*, 70–91. [CrossRef]
19. Li, C.; Wei, X.; Wang, J.; Wang, S.; Zhang, S. A Review of Reinforcement Learning Based Hyper-Heuristics. *PeerJ Comput. Sci.* **2024**, *10*, e2141. [CrossRef] [PubMed]
20. Slivkins, A. Introduction to Multi-Armed Bandits. *Found. Trends Mach. Learn.* **2019**, *12*, 1–286. [CrossRef]
21. Li, L.; Chu, W.; Langford, J.; Schapire, R.E. A Contextual-Bandit Approach to Personalized News Article Recommendation. In *Proceedings of the 19th International Conference on World Wide Web*; ACM: Raleigh, NC, USA, 2010; pp. 661–670.
22. Chu, W.; Li, L.; Reyzin, L.; Schapire, R.E. Contextual Bandits with Linear Payoff Functions. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics, Fort Lauderdale, FL, USA, 11–13 April 2011*; PMLR: Cambridge, MA, USA, 2011; Volume 15, pp. 208–214.
23. Wu, Q.; Iyer, N.; Wang, H. Learning Contextual Bandits in a Non-Stationary Environment. In *Proceedings of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, Ann Arbor, MI, USA, 8–12 July 2018*; Association for Computing Machinery: New York, NY, USA, 2018; pp. 495–504.
24. Russac, Y.; Vernade, C.; Cappé, O. Weighted Linear Bandits for Non-Stationary Environments. In *Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 8–14 December 2019*; Curran Associates, Inc.: Red Hook, NY, USA, 2019; Volume 32.
25. Liu, F.; Lee, J.; Shroff, N. A Change-Detection Based Framework for Piecewise-Stationary Multi-Armed Bandit Problem. In *Proceedings of the AAAI Conference on Artificial Intelligence*; Association for the Advancement of Artificial Intelligence: Washington, DC, USA, 2018; Volume 32. [CrossRef]
26. Austin, E.; Morgan, L.E. Detecting Changes and Anomalies in Nonstationary Contextual Bandits with an Application to Task Categorisation. *Inf. Sci.* **2025**, *717*, 122270. [CrossRef]
27. Hart, P.; Nilsson, N.; Raphael, B. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Trans. Syst. Sci. Cyber.* **1968**, *4*, 100–107. [CrossRef]