machines

MDPI

Article

Collaborative Scheduling Optimization of Staging and Transfer Operations for Mixed Unmanned Aerial Vehicle Fleets at Offshore Wind Power Substations

Wei Han [1], Xiangyu Liu [1], Xichao Su [2], Bing Wan [3], Fang Guo [1] and Changjiu Li [1,*]

[1] Aviation Foundation College, Naval Aviation University, Yantai 264001, China; hanwei70cn@tom.com or hanwei70cnau@163.com (W.H.); clear220221@126.com (X.L.); guofang575856@163.com (F.G.)
[2] School of Aviation Operation and Support, Naval Aviation University, Yantai 264001, China; suxich@126.com
[3] Coastal Defence College, Naval Aviation University, Yantai 264001, China; wanbing_0908@163.com
* Correspondence: jiu.haha@hrbeu.edu.cn

Abstract

Offshore wind power substations serve as fixed bases for unmanned aerial vehicle logistics distribution. In the mixed-fleet scheduling of multiple unmanned aerial vehicle types, parking spot allocation, configuration decisions, and transfer timing are deeply coupled, rendering traditional scheduling algorithms ineffective for efficient solution. To address this, this paper takes the context of substations performing material delivery and inspection missions to surrounding wind turbines, maintenance vessels, and other offshore facilities. An optimization model is constructed that comprehensively considers group priority, aircraft type differentiation, and configuration transition factors, with objectives of minimizing total transfer time, number of configuration changes, and workload variance among transfer crews. An event-driven configurable greedy initial solution generation strategy is introduced, and an adaptive large neighborhood search algorithm is designed, incorporating domain-knowledge-driven destruction and repair operators, an adaptive weight mechanism, and a simulated annealing acceptance criterion. Experimental results demonstrate that the proposed algorithm achieves the optimal mean objective values across all six test cases, outperforming the conventional adaptive large neighborhood search algorithm by 7.0–7.9% and the genetic algorithm by 14.4–22.1%, while maintaining the lowest standard deviation. The algorithm exhibits robust stability and practical applicability, providing effective decision support for unmanned aerial vehicle logistics distribution scheduling at offshore wind power substations.

Keywords: offshore wind power substation; unmanned aerial vehicle logistics distribution; scheduling optimization; configuration optimization; adaptive large neighborhood search

Academic Editor: Philip Moore

Received: 2 September 2026
Revised: 23 September 2026
Accepted: 23 September 2026
Published: 26 September 2026

1. Introduction

Offshore wind farms constitute a critical component of the renewable energy system, and their construction and operation costs directly determine the economic viability of wind power projects. As offshore wind power extends into deeper and more distant sea areas, the increasing distance from shore has rendered traditional vessel-based supply methods a prominent bottleneck constraining efficient wind farm operation, due to their susceptibility to sea conditions, long voyage times, and high costs. Unmanned aerial vehicles (UAVs), with their advantages of rapid direct access, flexible scheduling, and all-weather operability, are progressively emerging as a novel technical means for offshore wind power logistics support. In recent years, major offshore wind power nations have successively conducted application validations of UAVs in submarine cable inspection, wind turbine component transport, and emergency material delivery. As the core platform for power collection and operation and maintenance command within a wind farm, the offshore wind power substation, with its stable power supply, spacious operational space, and comprehensive communication conditions, is naturally suited to serve as a fixed base for UAV logistics distribution [1], which is shown in Figure 1. Against this backdrop, this paper conducts forward-looking modeling and optimization research on UAV fleet scheduling problems with the substation as a fixed base, targeting future routine operation scenarios.

(**a**) Offshore wind power substation. (**b**) UAV cargo route flight.

Figure 1. Offshore UAV logistics operations.

However, the UAV logistics operation workflow at offshore wind power substations encompasses multiple phases, including recovery and hangar storage, deck transfer and support, and takeoff and departure. These phases are deeply coupled in terms of parking spot occupancy, transfer equipment usage, and temporal coordination [2]. Multiple UAVs are required to sequentially complete a series of continuous procedures, including return landing, hangar recovery, cargo loading, battery charging inspection, and pre-departure staging. These procedures form complex temporal dependencies through shared resources such as elevators, tow tractors, and parking spots [3]. Simultaneously, the wing-folding configuration of UAVs directly affects parking spot compatibility and transfer path passability, causing configuration decisions, scheduling timing, and parking spot allocation to be deeply coupled with one another. The interplay of the above factors constitutes a complex combinatorial optimization problem characterized by multi-stage continuous operations, shared resource contention, and deeply coupled spatial and temporal constraints, posing severe challenges to both the expressive power of the scheduling model and the global search capability of the solution algorithm.

To address the above issues, the main contributions of this paper are as follows:

(1) An optimization model is constructed for the collaborative scheduling of staging and transfer operations for mixed UAV fleets preparing for re-launch at offshore wind power substations. The model comprehensively considers group priority, aircraft type differentiation, and configuration transition factors, with the objectives of minimizing total transfer time, the number of configuration changes, and the workload variance among transfer crews.

(2) An event-driven configurable greedy (EDCG) initial solution generation strategy is proposed. By means of an event-driven mechanism and configuration–space joint state maintenance, high-quality feasible initial solutions are generated.
(3) Domain-knowledge-driven destruction and repair operators are designed, and an adaptive large neighborhood search (ALNS) algorithm is constructed by combining an adaptive weight mechanism with a simulated annealing acceptance criterion, effectively balancing global exploration and local exploitation capabilities.
(4) The effectiveness of the proposed model and algorithm in terms of solution quality, stability, and computational efficiency is validated through multiple simulation instances of different scales, and comparative analyses are conducted with the conventional adaptive large neighborhood search algorithm and the genetic algorithm.

The research in this paper is based on the following assumptions:

(1) The deck layout and static obstacles remain unchanged throughout the scheduling horizon.
(2) Disruptions such as equipment failures and dynamic changes in task priorities are not considered, and once a task is initiated, it will not be interrupted until completion.
(3) The transfer process is assumed to be discrete in time and divided by a fixed time step, and the discrete time is sufficient to represent changes in the scheduling state.
(4) Communication delays, sensor errors, and actuator response delays are not considered.

This paper also has several limitations. The impact of unexpected events on scheduling strategies is not fully considered, and a complete rotation scheduling capability has not yet been formed. The model adopts deterministic parameters without fully considering offshore environmental uncertainties, and the algorithm is validated only through numerical simulations. These limitations are further discussed in Section 8.6.

The remainder of this paper is organized as follows. Section 2 reviews related research. Section 3 describes the problem, and Section 4 establishes the mathematical model. Section 5 introduces the preprocessing and key mechanisms, and Section 6 designs the EDGC-ALNS algorithm. Section 7 presents numerical simulations, Section 8 provides the discussion, and Section 9 concludes the paper.

2. Related Research

To ensure the comprehensiveness, representativeness, and timeliness of the literature review, a systematic retrieval and screening method was adopted. The literature sources mainly include Web of Science, IEEE Xplore, SpringerLink, CNKI, and MDPI. Three criteria were applied: topic relevance, publication level, and publication time. Priority was given to peer-reviewed journal articles published within the last five years that are closely related to UAV scheduling on fixed platforms, as well as studies involving adaptive large neighborhood search and its comparison algorithms.

Some scholars have conducted exploratory research on UAV scheduling on offshore platforms. For the recovery phase, early studies predominantly adopted static deterministic sequencing approaches, employing mixed-integer programming or heuristic rules to generate fixed recovery sequences. With growing awareness of uncertainties such as UAV go-arounds and energy consumption, Monte Carlo simulation and stochastic programming methods were subsequently introduced [4], extending the problem to dynamic decision-making under stochastic disturbances. In recent years, the integration of deep reinforcement learning and long short-term memory networks has further evolved recovery sequencing into a sequential decision-making problem under partial observability, achieving a transition from offline static optimization to millisecond-level real-time resequencing [5,6].

Research on the operational area transfer and support phase has undergone a paradigm shift from "multi-station" support to "one-stop" support. Early models predominantly adopted mixed flow shop or flexible job shop scheduling frameworks [7], treating UAVs as jobs with fixed processing sequences. As the demand for modeling fidelity increased, resource-constrained project scheduling (RCPS)-type models [8] gradually became mainstream. Such models can describe the network-structured execution order of support tasks while also accommodating realistic constraints such as transfer times and coverage ranges of support personnel and equipment across different workstations [9], thereby rendering the mathematical formulation more closely aligned with the problem's intrinsic characteristics. Concurrently, the research perspective has expanded from scheduling optimization under fixed resource configurations to the joint optimization of resource station configuration and scheduling schemes [10], revealing the marginal effect between incremental resource availability and makespan reduction. In recent years, deep reinforcement learning methods have been progressively introduced into this domain. By encoding task precedence dependencies via graph neural networks, decision-making time has been compressed from the minute-level to the second-level [11]. Furthermore, some studies have incorporated variable job-shop structures into the model [12], relaxing the conventional assumption of fixed task sequences and imparting greater flexibility to the scheduling model.

The research on the departure and sortie phase has evolved from single-UAV scheduling to multi-UAV coordinated scheduling. Early studies mostly adopted metaheuristic methods such as genetic algorithms and particle swarm optimization. As spatial constraints became better understood, factors including taxing collision avoidance, wake separation, and minimum sortie interval were gradually incorporated, shifting the problem from temporal optimization to spatiotemporally coupled scheduling [13,14]. In terms of solution methods, the introduction of constraint programming and mixed-integer programming techniques improved the solution efficiency for small- and medium-scale problems and provided a certain online planning capability. For sortie scheduling under towing mode, hybrid flow-shop modeling and chaos-initialized genetic algorithms were successively proposed, validating the superiority of the towing sortie strategy [14]. Meanwhile, discrete event simulation methods were used to quantitatively evaluate the relationship among deck layout, number of carrier aircraft, and sortie rate, providing a statistically meaningful validation means for scheduling schemes [15]. In recent years, virtual-physical fusion and hardware-in-the-loop technologies have been introduced into sortie scheduling research, enabling semi-physical validation and real-time visualization of scheduling algorithms through digital twin platforms [16]. Small-sample graph neural networks and related methods have been applied to predict the sortie mission reliability of shipborne vehicle layouts, achieving higher prediction accuracy and computational efficiency than traditional reliability calculation methods under limited sample conditions [17]. These studies indicate that sortie scheduling is developing from single heuristic solving toward a combination of multi-constraint coupled modeling, simulation validation, and intelligent prediction. The methods and models adopted in the relevant literature are summarized in Table 1.

Table 1. Current research status of operational area operations.

Phase	Author(s)	Year	Model	Solution Algorithm
Recovery and landing	Chagas et al. [4]	2025	Markov decision process with chance constraints	Cost function approximation policy
	Han et al. [6]	2026	Partially observable Markov decision process	DRL
Operational area support	Cui et al. [8]	2021	Resource-constrained multi-project scheduling	Dual-population multi-operator genetic algorithm
	Liu et al. [10]	2023	Bi-level optimization model	Improved variable neighborhood search
	Chen et al. [11]	2025	Markov decision process	DRL
	Guo et al. [12]	2025	Extended resource-constrained multi-project scheduling	Improved particle swarm optimization
	Su et al. [9]	2026	Markov decision process	DRL
Launch and departure	Jia et al. [13]	2026	Multi-constraint mission planning model	Self-search differential evolution algorithm
	Deng et al. [14]	2025	Hybrid flow-shop scheduling	Chaos-initialized genetic algorithm
	Yoon et al. [15]	2025	Discrete event simulation model	Genetic algorithm

While the aforementioned studies have achieved notable advances within their respective phases, in actual scheduling scenarios, UAVs require continuous transfer across multiple phases, rendering the transfer tasks highly coupled and subject to numerous constraints. Taking the post-recovery staging transition problem investigated in this paper as an example, UAVs must sequentially undergo recovery positioning, support operations, and pre-departure staging. These phases are interconnected by shared resources such as tow tractors and elevators, forming intricate temporal dependencies. Conventional phase-decoupled optimization methods are inadequate for handling cross-phase resource competition and temporal coordination, thereby necessitating an integrated, end-to-end scheduling approach.

Since its inception, the ALNS algorithm has been widely applied in various combinatorial optimization domains, including vehicle routing, production scheduling, and port operations, owing to its flexible destroy-and-repair framework and strong adaptability to complex constraints [18]. Researchers have proposed multiple enhancements to ALNS tailored to the structural characteristics of different problems: in terms of hybridization strategies, mechanisms such as tabu search [19] and simulated annealing [20] have been integrated to compensate for local search deficiencies—for instance, Lei et al. [21] hybridized ALNS with both tabu search and simulated annealing for electric vehicle routing with delivery options. In operator design, generic operators have been progressively replaced by customized operators embedding domain knowledge [22]. In selection mechanisms, adaptive weight strategies [23] have been continuously refined, and more recently, deep reinforcement learning [24,25] and Q-learning have been incorporated to enable intelligent configuration of operators and parameters [26]. These advances provide valuable references for designing an ALNS algorithm tailored to the post-recovery UAV fleet staging transition problem. Table 2 summarizes the key features of representative improvement studies.

Table 2. Summarizes the key features of representative ALNS enhancements.

Author	Model Improvement	Destruction Operator	Repair Operator	Acceptance Criterion
Koyuncuoğlu and Demir [20] (2021)	ALNS combined with initialization strategy based on the storage bowl phenomenon	Position-based removal-insertion, machine-state-based removal-insertion	Paired design with destruction operators	Simulated annealing
Gao et al. [19] (2023)	Hybrid of ALNS and tabu search	Random removal, worst-distance removal, distance-similarity removal, etc.	Random insertion, regret insertion, greedy insertion, etc.	Simulated annealing
Mao et al. [22] (2025)	Domain knowledge-driven initial solution and operator design	Maximum conflict removal, maximum observation opportunity removal, maximum resource-switch removal, etc.	Minimum conflict insertion, minimum observation opportunity insertion, minimum resource-switch insertion, etc.	Metropolis criterion
Lei et al. [21] (2025)	Triple hybrid of ALNS, tabu search, and simulated annealing	Random deletion, maximum correlation removal, worst-cost removal, etc.	FIFO greedy insertion, regret-value insertion, etc.	Simulated annealing
Li et al. [23] (2025)	UAV-truck path with arbitrary dynamic take-off/landing positions	Random destruction, most-critical destruction, correlation destruction	Random repair, greedy repair, regret-value repair	Simulated annealing
Chen et al. [25] (2025)	DRL for operator selection and parameter configuration	Random destruction, minimum-gain destruction, maximum-opportunity destruction, etc.	Maximum-gain repair, greedy repair, minimum-opportunity repair, etc.	DRL dynamically controls the simulated annealing temperature

These improvements have enhanced the performance of ALNS from multiple perspectives. At the hybridization level, Gao et al. [19] combined ALNS with tabu search, leveraging the short-term memory mechanism of the tabu list to effectively avoid redundant search, thereby significantly accelerating convergence in maintenance task scheduling. At the operator design level, Mao et al. [22] addressed low-orbit early warning satellite scheduling by introducing conflict degree and observation opportunity-driven destruction operators, rendering the neighborhood search more problem-specific. At the parameter control level, Chen et al. [25] embedded deep reinforcement learning into the ALNS framework to enable intelligent configuration of operator selection and temperature control, substantially reducing manual tuning efforts. These enhancements indicate that incorporating problem-specific domain knowledge into initial solution construction, operator design, and parameter adaptation constitutes an effective pathway for improving algorithmic adaptability and efficiency. Furthermore, ALNS has been extended to address combinatorial problems under uncertainty; Keskin et al. [27] integrated ALNS with discrete-event simulation in a two-stage framework to solve electric vehicle routing problems with stochastic charging waiting times, demonstrating the algorithm's robustness in uncertain environments.

However, the aforementioned studies predominantly address single-phase or loosely coupled scheduling problems. In contrast, the UAV re-sortie problem investigated in this paper is characterized by multi-stage continuous operations, shared resource contention, and deeply coupled spatial and temporal constraints, rendering existing ALNS frameworks unsuitable for direct application. First, UAVs form cross-phase resource dependencies through tow tractors and elevators across recovery positioning, support operations, and pre-departure staging; the destroy and repair operators must therefore preserve the temporal consistency of operations across all phases. Second, the compatibility of parking spots is mutually dependent on the wing-folding/unfolding configuration state of each UAV, and the update of the available parking spot set induced by configuration changes must be explicitly represented within the operator design. Consequently, a customized ALNS solution framework that accommodates these problem-specific features is required.

Despite the significant progress achieved by existing studies in both modeling fidelity and solution efficiency, two major gaps persist in the current literature on offshore UAV operations. First, in operational practice, UAVs are typically deployed in mixed formations, where group priority imposes hard constraints on the sequencing of support tasks. Moreover, the differing geometric envelopes of various UAV types directly affect parking spot compatibility and spatial allocation schemes. However, existing models inadequately capture such group–space coupling constraints. Second, UAV configuration—as an actively adjustable variable influencing spatial occupancy—has not yet been incorporated into the scheduling optimization framework. Most studies assume fixed configurations and fail to leverage configuration transitions to improve spatial utilization of the operational area under spatial congestion.

To address these gaps, this paper develops an optimization model that comprehensively accounts for group priority, aircraft type differentiation, and configuration transition factors. The model is formulated to minimize total transfer time, number of configuration changes, and workload variance among transfer crews, and is solved using an ALNS algorithm. Problem-specific destruction and repair operators are designed, and an adaptive weight mechanism together with a simulated annealing acceptance criterion is introduced to enhance the algorithm's capability to escape local optima. The effectiveness of the proposed model and algorithm is validated through multiple simulation instances of varying scales.

3. Problem Description

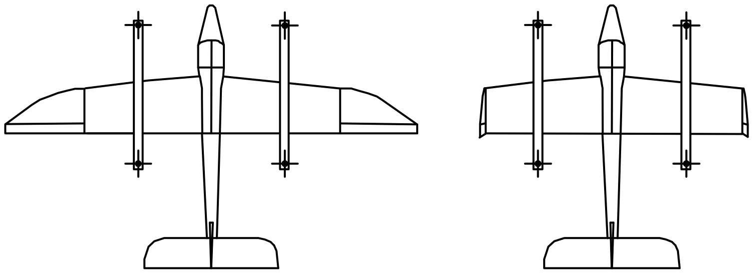
3.1. Re-Sortie-Oriented Recovery and Support Workflow for UAVs at Offshore Substations

Following the recovery of the final wave of UAVs at the offshore substation, the fleet undergoes pre-departure turnaround support, comprising transfer of malfunctioning or non-sortie UAVs to the hangar, retrieval of newly tasked UAVs from the hangar, and sequential servicing of re-sortie-designated UAVs. Priority delivery groups are then marshalled to pre-departure parking spots according to the launch sequence of operational groups, with remaining UAVs dynamically positioned based on platform availability and configuration constraints. This multi-stage process features distinct operational contents, resource demands, and spatial constraints across different phases. Specifically, the multi-stage support operations can be divided into the following three phases:

(1) Recovery and hangar-stowage phase. Recovery order is determined by delivery-group and intra-group priorities. Landing intervals are set based on parking-spot coincidence or interference to ensure safety, with recovery parking decisions made under temporary-spot capacity constraints. Fixed-wing UAVs are towed to temporary parking spots; multi-rotor UAVs requiring re-sortie are routed directly to servicing spots, while malfunctioning or non-sortie multi-rotor UAVs are elevator-transferred to the hangar. Post-recovery, malfunctioning or non-follow-on fixed-wing UAVs are also lowered to the hangar and stowed per available hangar-bay spots, all in folded-wing configuration.

(2) Servicing phase. Aircraft with mission requirements of various types are transferred from transient spots or the hangar to servicing spots, where servicing deployment decisions are executed and one-stop turnaround servicing is performed at these positions. Upon arrival of UAVs at the servicing spots, a payload mounting type

assessment is conducted. If no payload mounting is required, other servicing tasks are executed; if payload mounting is required, the mounting position is further determined: for belly-mounted payload, the current wing configuration is retained, whereas for wingtip-mounted payload, reconfiguration to the spread-wing state is required, as shown in Figure 2.



(**a**) Wing-Extended Configuration. (**b**) Wing-Folded Configuration.

Figure 2. Comparison of Different UAV Configurations.

(3) Pre-departure staging phase. To facilitate takeoff in group sequence during re-sortie operations, the principle of "higher-priority delivery groups serviced and staged first" shall be followed: higher-priority delivery groups are prioritized for servicing and are then transferred to pre-departure parking spots for staging. Among them, fixed-wing UAVs are reconfigured to the spread-wing state after transfer to the pre-departure spots to facilitate pre-flight inspection, while multi-rotor UAVs remain folded and will be spread before launch. If pre-departure parking space is insufficient, UAVs hold at servicing spots in their current configuration until space becomes available for subsequent transfer.

To clearly illustrate the scheduling relationships among the offshore substation's functional zones and the flow paths of the three UAV categories—recovered-and-re-sortie, newly sortied from the hangar, and recovered-and-stored, Figure 3 depicts the layout of the flight operational area and hangar zones along with the corresponding scheduling flow. The red, blue, and yellow aircraft flow directions in the figure represent the transfer flows of three types of aircraft: recovery followed by re-sortie, recovery followed by hangar storage, and new sortie from the hangar, respectively.

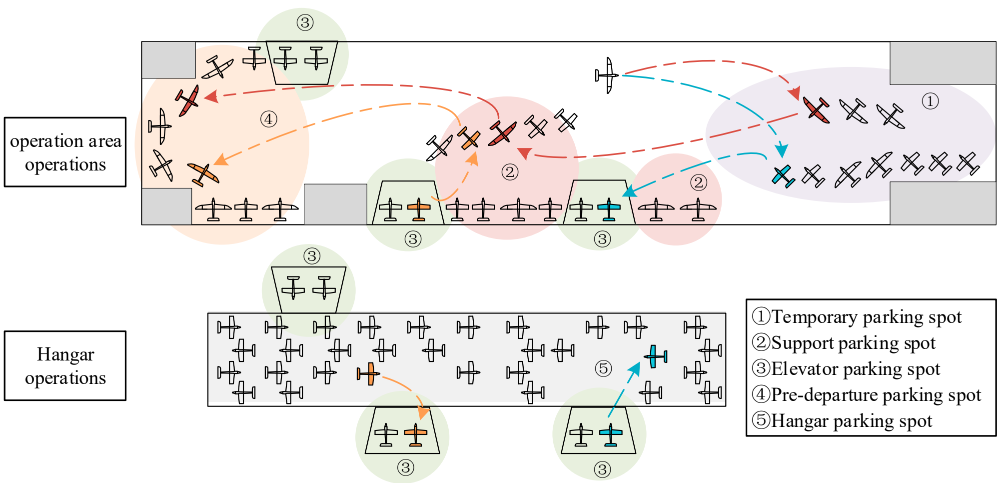


Figure 3. Schematic diagram of operational area functional zones and aircraft re-sortie preparation workflow.

The re-sortie UAV group comprises two components: UAVs from the recovered wave that are designated for re-sortie, and newly sorted UAVs retrieved from the hangar. The former are first transferred to temporary parking spots and then to servicing spots after all recovery operations are completed. The latter are transferred via elevators from hangar parking spots to platform servicing spots. After support, both categories are either transferred to pre-departure parking spots or remain in place, depending on available space. Recovered UAVs that are faulty or have no follow-on missions are transferred via elevators to the hangar immediately after recovery.

To provide a clear depiction of the flow paths of the three aircraft categories and the operational and decision-making content at each phase, Figure 4 presents the Re-sortie-oriented recovery support flowchart for UAVs at offshore substations.

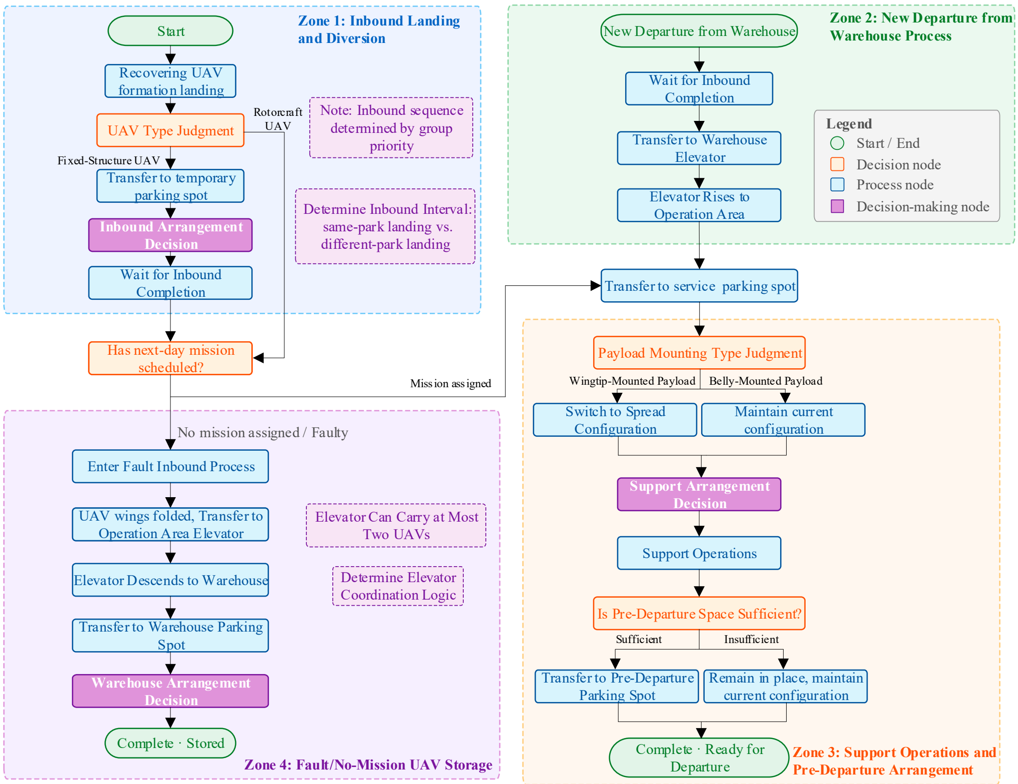


Figure 4. Re-sortie-oriented recovery support flowchart for UAVs at offshore substations.

3.2. Problem Definition

The problem is formally defined as follows: Given a mixed UAV fleet consisting of multiple delivery groups, each UAV's recovery group and delivery group affiliations, along with their respective priorities, are known. Based on UAV status, the scheduling decisions must coordinate the recovery and storage of the recovered wave as well as the retrieval and support of re-sortie UAVs. For each UAV at each phase, decisions regarding parking spot assignment, transfer timing, and UAV configuration must be made in a coordinated manner. Each UAV is transferred from its initial parking spot to its destination parking spot along a pre-specified feasible path selected from a pre-constructed path library, which already satisfies single-UAV kinematic constraints and static obstacle avoidance requirements.

At the start and end of each transfer operation, the UAV releases and occupies parking spot resources, respectively. Equipment resources required during transfer include platform-area tow tractors, hangar tow tractors, and elevators, with the provision that platform-area and hangar tow tractors are not permitted to operate across zones. Accordingly, each transfer operation entails the coordinated decision of the following key elements: destination parking spot assignment, equipment resource allocation (including tow tractors and elevators), transfer start time, and UAV configuration upon transfer completion. These decision variables are mutually coupled and collectively constitute a complete UAV transfer scheduling solution. In determining the above decisions, a series of complex constraints must be satisfied, encompassing fundamental resource allocation, concurrent operation restrictions, logical temporal dependencies, and dynamic collision safety requirements. These constraints exhibit pronounced hierarchical structure and coupling characteristics. Under this constraint framework, the quality of a candidate solution is evaluated by three optimization objectives: minimizing the total transfer time, minimizing the number of configuration changes, and balancing equipment workload.

The core challenges of this problem manifest in the following three aspects. First, strong coupling among configuration, space, and timing: Configuration transition decisions directly alter the set of available parking spots, update spatial occupancy states, and activate or deactivate the validity conditions of temporal constraints; these three dimensions are tightly intertwined and cannot be decoupled independently. Second, spatiotemporal coupling in collision detection and avoidance: Collision safety during transfer operations must be verified in continuous spatiotemporal domains. Collision risk is closely related to the transfer start time, path selection, and motion speed of each UAV. Adjusting the start time or path of one UAV may eliminate a collision with another while simultaneously inducing new conflicts, exhibiting a trade-off coupling effect. Third, hierarchical constraints and conflict propagation: Violation of a single constraint is typically caused by multiple decision variables collectively, and rectifying one constraint may trigger secondary violations of others. This necessitates an algorithm capable of systemically handling constraint conflicts rather than repairing them in isolation.

4. Model Formulation

This section develops a scheduling model for re-sortie-oriented recovery support and staging transition of mixed UAV fleets at offshore substations. The following assumptions are adopted: any transfer operation, once initiated, cannot be interrupted; the transfer times derived from the pre-constructed path library are constant and independent of other simultaneously ongoing transfer operations; and disruptions due to unforeseen contingencies, such as equipment failure, are not considered.

4.1. Parameter Defination and Decision Variable

The model parameters are listed in Table 3.

Table 3. Parameter definitions.

Variable	Definition
I	Set of aircraft involved in transfer operations, where $I=\{1,2,\cdots,Nf\}$, $I=I_{cd}\cup I_{hs}$, I_{cd}, I_{hs} denote the launch group, recovery group, and the combined group, respectively
I_w	Set of aircraft assigned to mission w, $\bigcup_{w=1}^{Nw} I_w = I_{cd}, I_w \cap I_{w'} = \varnothing (w \neq w')$
q_w	Priority index of formation w, where a smaller value indicates a higher priority
Jz_i	Set of transfer tasks for aircraft i, $Jz_i=\{1,2,\cdots,Nz_i\}$
$Jz_i{}'$	Extended set of transfer tasks for aircraft i, where $Jz_i{}' = Jz_i \cup \{Nz_i+1\}$, with the last entry being a dummy termination task
Nz_i	Number of transfer stages for aircraft i, where $Nz_i \le 3$
O_{ij}	The transfer operation j of aircraft i
P_{ij}	Set of available parking spots for aircraft i at stage j

Q_{ij}	Parking spot where aircraft *i* is located prior to the transfer at stage *j*
M_{ij}	Set of available material handling equipment for the transfer of aircraft *k* at stage *j*
Kz	Set of material handling team categories
Lz_k	Set of material handling teams of category *k*, where $Lz_k = \{1, 2, \cdots, \lvert Lz_k \rvert\}$
z_{ijk}	=1 if team O_{ij} belongs to category *k*, and 0 otherwise
Cz_{ijeg}^{k}	Travel time of equipment *k* from the completion of operation O_{ij} to the start of operation O_{eg}
tx	Mooring/unmooring time
te	One-way transfer time of the elevator
ε	An arbitrarily small positive real number
δ_e	Minimum time interval between successive recoveries
BM	A sufficiently large positive real number
Wz	Set of aircraft operational groups, where $Wz = \{1, 2, \cdots, Nw\}$
c_{ij}	=1 denotes wings-spread configuration prior to O_{ij}, = 0 denotes wings-folded configuration. $j = Nz_i + 1$ indicates the final configuration of the aircraft
H	Set of aircraft servicing tasks involved in transfer operations, where $H = \{H_1, H_2, \cdots, H_{Nf}\}$
h_i	Payload mounting scheme for fixed-wing aircraft *i*; $h_i = 1$ denotes wingtip-mounted payload, $h_i = 0$ denotes belly-mounted payload

The decision variables are listed in Table 4.

Table 4. Decision Variables.

Variable	Definition
Sz_{ij}	Start time of transfer operation O_{ij}
Ez_{ij}	Completion time of transfer operation O_{ij}
A_{ijp}	=1: Aircraft *i* is parked at parking spot *p* prior to the transfer at stage *j* =0: Otherwise
Xz_{ijl}	=1: Transfer operation O_{ij} is handled by the $l\left(l \in M_{ij}\right)$ material handling equipment =0: Otherwise
Yz_{ijeg}	=1: Operation O_{ij} and operation O_{eg} occupy the same material handling equipment in the same transfer stage, and O_{ij} takes priority over =0: Otherwise
Ze_{wx}	=1: I_w and I_x are assigned to the same recovery spot, and I_w takes priority in the recovery sequence =0: Otherwise
$y_{w_1 w_2}$	=1: $q_{w_1} > q_{w_2}$ =0: Otherwise
d_{ij}	=1: Aircraft configuration changes during transfer operation O_{ij} =0: Otherwise

4.2. Objective Functions

To address the support requirements of mixed UAV fleet entry/exit from the hangar and platform transfer operations at offshore wind power substations, three optimization objectives are respectively formulated: minimization of total mission makespan, minimization of the number of configuration changes across the UAV fleet, and optimization of load balancing among transfer crews. Decisions are to be made regarding parking spot selection, UAV configuration states, and transfer timing at each support phase.

(1) Minimization of total mission completion time

The total mission completion time is defined as the maximum completion time among all unmanned aerial vehicles, as shown in Equation (1):

$$\min F_1 = \max_{i \in I} Ez_{i(Nz_i+1)} \tag{1}$$

(2) Minimization of total number of configuration changes

Changes in UAV configuration accelerate fatigue of wing-folding mechanisms, affect equipment service life and support efficiency, and introduce safety risks. Under the premise of satisfying support requirements, the number of configuration switchovers per UAV throughout the entire process should be minimized to enhance efficiency and reduce wear. This metric can be expressed as Equation (2):

$$\min F_2 = \sum_{i \in I} \sum_{j \in Jz_i{}'} d_{ij} \tag{2}$$

(3) Minimization of workload variance among transfer crews

In actual transfer support operations, each set of transfer equipment is assigned a transfer crew, responsible for tow tractor operation, full-process safety monitoring, and mooring/unmooring tasks. As the actual execution unit of transfer tasks, the load balancing problem of transfer crews is primarily discussed for two operational scenarios: the platform operational area and the hangar. To quantify the degree of workload balance among crews, the load variance and LBT are adopted as the metrics for workload balancing, the load variance among transfer crews is calculated as shown in Equation (3):

$$\min F_3 = \min LBT = \sum_{k \in Kz} \gamma_k \cdot \frac{\sum_{l \in Lz_k} \left(TB_{kl} - \overline{TB_k} \right)^2}{\left| Lz_k \right|} \tag{3}$$

where γ_k denotes the weight coefficient for the load variance of the transfer crew category $k(k \in Kz)$, and the workload of the transfer crew $l(l \in Lzk)$ in category $k(k \in Kz)$ is expressed as the weighted sum of aircraft towing operation time and transfer support time, as shown in Equation (4):

$$TB_{kl} = \sum_{i \in I} \sum_{j \in Jz_i} TR\left(Q_{ij}, Q_{i(j+1)} \right) \cdot Xz_{ijl} \cdot z_{ijk} + \omega_{\mathrm{lb}}^{\mathrm{k}} \cdot \sum_{i \in I} \sum_{j \in Jz_i} \sum_{e \in I} \sum_{g \in Jz_e} Y_{ijeg} \cdot Cp_{ijeg}^{k} \tag{4}$$

In the formula, $TR\left(Q_{ij}, Q_{i(j+1)} \right)$ is an abbreviation for $TR\left(Q_{ij}, Q_{i(j+1)}, Ap_{Sz_{ij}}\left(Q_{ij}, Q_{i(j+1)} \right) \right)$, denoting the transfer time from parking spot Q_{ij} to $Q_{i(j+1)}$ at time Sz_{ij}; $\omega_{\mathrm{lb}}^{\mathrm{k}}$ denotes the load weight of the transfer time of category $k(k \in Kz)$ transfer crews relative to the towing operation time; and the mean load of category k transfer crews is $\overline{TB_k}$.

(4) Comprehensive objective function

After normalizing the three sub-objectives to balance dimensional differences, the linear weighted sum method is adopted to aggregate them into a single objective function, as shown in Equation (5):

$$\min F = \min\left(\lambda_1 \cdot \tilde{F}_1 + \lambda_2 \cdot \tilde{F}_2 + \lambda_3 \cdot \tilde{F}_3 \right) \tag{5}$$

where $\tilde{F}_i = \dfrac{F_i}{N_i}$ denotes the normalized sub-objective function, N_i is the corresponding normalization benchmark value, $\lambda_1, \lambda_2, \lambda_3$ are the weight coefficients of the three sub-objectives, respectively, and $\lambda_1 + \lambda_2 + \lambda_3 = 1$.

The importance levels of the sub-objectives are ranked as "total transfer time > number of configuration changes > load variance." Total transfer time is the primary indicator of platform scheduling efficiency and is assigned the highest weight. The number of configuration changes affects the service life of wing-folding mechanisms and operational safety, and is assigned the second-highest weight. Load balancing reflects resource utilization and operational fairness, and is assigned the baseline weight. The final weights are determined through the Pareto sensitivity analysis in Section 7.3.

4.3. Constraints

(1) Resource allocation and capacity constraints

During any transfer operation phase, each UAV can be assigned to at most one piece of equipment responsible for transfer operations in that phase, and each piece of transfer equipment can tow at most one UAV at a time. Prior to the start of each transfer phase, each aircraft must be assigned to exactly one parking spot, and each parking spot can accommodate at most one aircraft. Constraints can be expressed as Equations (6) and (7):

$$\sum_{l \in M_{ij}} Xz_{ijl} = 1, \sum_{p \in P} A_{ijp} = 1, \forall i \in I, \forall j \in Jz_i \tag{6}$$

$$\sum_{i \in I} A_{ijp} \leq 1, \forall j \in Jz_i, \forall p \in P_{ij} \tag{7}$$

(2) Launch/recovery sequencing constraints

Due to the influence of downwash airflow, a time interval should be maintained between two successively recovered UAVs. Different groups are recovered on a first-come-first-served basis, while within the same group, the landing sequence is determined by coding priority, as shown in Equation (8):

$$Sz_{ej} \geq Ze_{wx}\left(Ez_{ij} + \delta_e\right), \forall i \in I_w, \forall e \in I_x, j = 1 \tag{8}$$

(3) UAVs transfer sequencing constraints

The transfer sequencing constraints of UAVs include three aspects: first, the start and end time constraints of transfer operations in each phase, and the switch time constraints for operations across different phases, as shown in Equations (9) and (10):

$$Ez_{ij} = Sz_{ij} + TR\left(Q_{ij}, Q_{i(j+1)}\right) + tx, \forall i \in I, \forall j \in Jz_i \tag{9}$$

$$Sz_{i(j+1)} \geq Ez_{ij} + tx, \forall i \in I, \forall j \in \{1,2\} \tag{10}$$

Second, the temporal constraints for entry/exit position transfer conflict avoidance. The start time of any transfer task must be later than the start times of all transfer tasks that occupy its path; the completion time of any transfer task must be earlier than the completion times of all other tasks that occupy its parking-in path, as shown in Equations (11) and (12):

$$Sz_{ij} \geq \max_{i' \in I_s}\left(Sz_{i'j} + \varepsilon\right) \tag{11}$$

$$Ez_{ij} \leq \min_{i^* \in I_e}\left(Ez_{i^*j'} - \varepsilon\right) \tag{12}$$

Among them, I_s and I_e denote the sets of conflicting aircraft for the parking-in and parking-out operations of aircraft i in phase j, respectively.

Third, a UAV can commence transfer only after the assigned towing equipment has arrived at its position; thus, the transfer start time must be no earlier than the equipment arrival time, as shown in Equation (13):

$$Sz_{eg} \geq Ez_{ij} + Cz_{ijeg}^{k} + tx \tag{13}$$

(4) Delivery group support sequencing constraints.

Different UAV groups undertake operational time requirements of varying levels; groups with higher launch priority shall be given priority in completing support operations and be positioned at the pre-departure parking spots, so as to facilitate rapid pre-launch preparations for subsequent re-sorties, as shown in Equation (14):

$$E_{ij}^{s} \le E_{eg}^{s} + BM \cdot \left(1 - y_{w_1 w_2}\right), \forall i \in I_{w_1}, \forall e \in I_{w_2}, \forall j \in J_i, \forall g \in J_e \quad (14)$$

Among them, E_{ij}^{s} denotes the completion time of support operations for aircraft i in phase j.

(5) Transfer equipment repositioning support sequencing constraints

After completing its preceding towing operation, the transfer equipment must be repositioned to the parking spot of the next aircraft to be transferred, before a new transfer operation can be initiated; as shown in Equation (15):

$$Ez_{ij} + Cz_{ijeg}^{k} \cdot Yz_{ijeg} \le Sz_{eg} + BM \cdot \left(1 - Yz_{ijeg}\right), \forall k \in \{1,2,3\}, \forall i,e \in I, \forall j \in Jz_i, \forall g \in Jz_g \quad (15)$$

(6) Fleet transfer collision avoidance constraints

The safety of the fleet transfer process is the most fundamental and important constraint, primarily reflected in collision avoidance safety. Collision avoidance safety requires that at any given time, the spatial positions of any two aircraft do not overlap. For any time $t \in [Sz_{i1}, Ez_{iNz_i}]$, let the spatial region occupied by the UAV i be Sp_t^i; then the multi-UAV transfer collision avoidance constraint can be expressed as Equation (16):

$$Sp_t^i \cap Sp_t^e = \varnothing, \forall i,e \in I, t \in [Sz_{i1}, Ez_{iNz_i}] \cap [Sz_{e1}, Ez_{eNz_e}] \quad (16)$$

(7) Configuration and operation matching constraints

When UAVs undergo support operations, certain procedures have specific requirements for UAV configuration. For UAVs undertaking payload mounting tasks, when wingtip-mounted payload is mounted, the UAV configuration must remain in the wings-spread state; when belly-mounted payload is mounted, the UAV may maintain its current configuration. UAVs without tasks or with faults are required to be stored in the hangar, and thus their wings are folded immediately after landing. The constraints can be expressed as Equations (17) and (18):

$$c_{ij} \cdot h_i = 1, \forall i \in I, j = Jz_i, Mg \in H_i \quad (17)$$

$$c_{ij} = 0, \forall i \in I_{hs} \setminus I_{cd}, j \in \{2, Jz_i + 1\}, \quad (18)$$

where Mg denotes the set of payload mounting tasks.

In addition, the final configuration states of UAVs with different flow directions are also fixed. UAVs without tasks or with faults that are ultimately parked in the hangar or on the platform are required to have wings folded to reduce spatial occupancy; UAVs for re-sortie or new sortie are both required to have wings spread for wing inspection prior to takeoff. The constraints can be expressed as Equations (19) and (20):

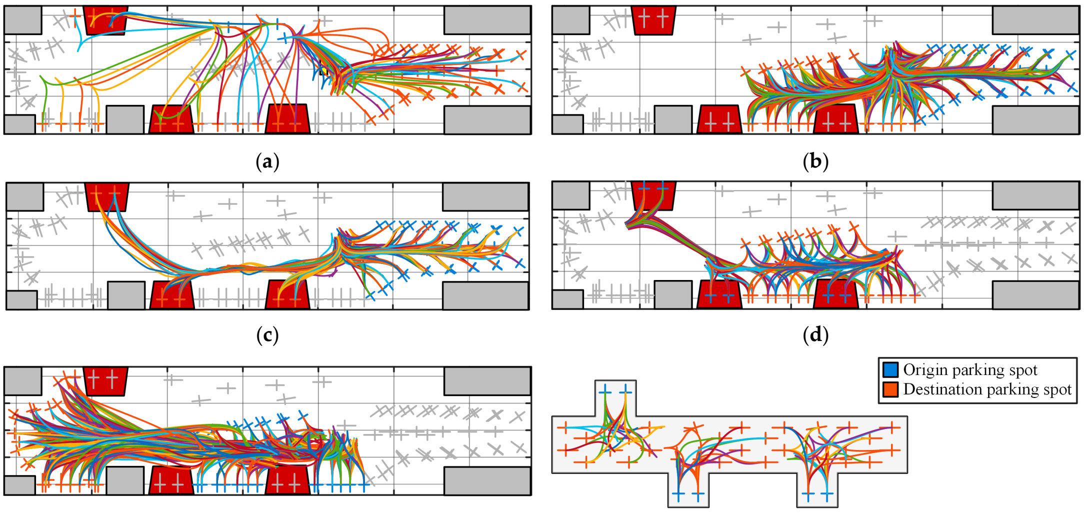
$$c_{ij} = 0, j = Nz_i + 1, \forall i \in I_{cd} \quad (19)$$

$$c_{ij} = 1, j = Nz_i + 1, \forall i \in I_{hs} \quad (20)$$

5. Preprocessing and Key Mechanism Analysis

5.1. Path Library Construction

The platform operations for re-sortie of mixed UAV fleets encompass multiple types of transfer tasks, and real-time online path solving during scheduling is computationally expensive. To address this, a path library covering typical transfer tasks is pre-constructed to decouple offline path planning from online scheduling decisions. The hybrid A* algorithm is employed to generate collision-free smooth trajectories satisfying kinematic constraints and terminal pose requirements under known static obstacle conditions [28,29]. In the trajectory generation process, the safety margin is determined according to the radius of the aircraft characteristic circle, and one-tenth of this radius is adopted as the safety distance constraint, thereby balancing path feasibility and transfer safety. Figure 5 presents a schematic diagram of the path library construction, illustrating the primary transfer paths among temporary parking spots, one-stop support spots, elevator parking spots, and stern parking spots, as well as the hangar transfer paths.



(a) (b) (c) (d) (e) (f)

Figure 5. Schematic diagram of the path library composition for the operation phase. (**a**) Inbound Parking Spot—Temporary Parking Spot/Support Parking Spot/Operation Area Elevator. (**b**) Temporary Parking Spot—Support Parking Spot. (**c**) Temporary Parking Spot—Operation Area Elevator. (**d**) Operation Area Elevator—Support Parking Spot. (**e**) Support Parking Spot—Pre-Departure Parking Spot. (**f**) Warehouse Elevator—Warehouse Parking Spot.

5.2. Elevator Coordination Logic

The elevator is a critical transfer node connecting the platform operational area and the hangar, and its allocation strategy directly affects the temporal efficiency of multi-UAV transfer operations. When multiple UAVs perform inter-level transfers on the same elevator, a coordination mechanism must be established to avoid resource conflicts.

Assume that UAV A is currently at the elevator parking spot and is about to commence an inter-level transfer. The system first checks whether any other UAV is currently occupying or has pre-occupied the elevator. If not, A proceeds as originally scheduled. Otherwise, let the current phase start times of UAV A and B be t_A^s and t_B^s, respectively, and let the current time be t. The coordination logic is illustrated in Figure 6.

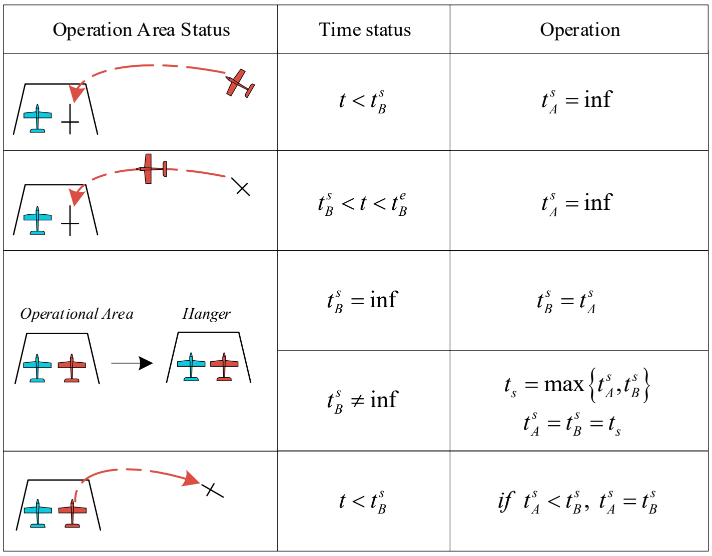


Figure 6. Elevator coordination logic.

In the figure, the blue and red UAVs represent UAV A and B, respectively. Based on the phase and start time status of the other UAV, the decision rules are as follows: if B is in the phase of entering the elevator, A must wait until B is in position before both commence the transfer together; if both UAVs are in the inter-level transfer phase, the UAV that started earlier waits for the UAV that started later; if B is in the phase of leaving the elevator, A must wait until B has completely cleared the elevator area before starting. Through the above coordination rules, safe and orderly sharing of elevator resources is achieved, avoiding temporal conflicts and operational delays caused by multiple UAVs contending for the same elevator.

6. EDGC-ALNS Algorithm Design

6.1. Initial Solution Generation

The quality of the initial solution significantly affects the convergence speed and final solution quality of the ALNS algorithm. This paper proposes an EDCG strategy to generate an initial feasible solution.

Given that the configuration state of UAV *i* in phase *j* is $c_{ij} \in \{0,1\}$, where 1 denotes wings spread and 0 denotes wings folded. The configuration state space *C* and the parking spot state space *S* are coupled through an interference matrix, forming the configuration-space joint state space $\Omega = C \times S$. When UAV is parked at parking spot *s* in configuration *c*, the spatial region it occupies is $A(s,c)$, and any two UAV *i* and *e* satisfy the collision avoidance constraint $A(s_i,c_i) \cap A(s_e,c_e) = \varnothing$. The EDCG strategy maintains the real-time projection of Ω during the scheduling process, ensuring collaborative consistency between configuration decisions and spatial allocation.

The scheduler adopts an event-driven mechanism, processing three types of events in chronological order: transfer start, transfer end, and support completion. Immediately after event processing is completed, Ω is updated, and state propagation is executed along the constraint propagation network $G_{cons} = (V, E_B \cup E_G)$ to update the blocking status of all affected UAV. Subsequently, the parking spot selection and resource allocation functions are invoked from the set of schedulable UAV according to priority to complete the decision-making process. Here, *V* denotes the set of parking spot nodes, E_B denotes the parking-out constraint edges, and E_G denotes the parking-in constraint edges.

The scheduling scheme generation procedure of the EDCG strategy is illustrated in Figure 7.

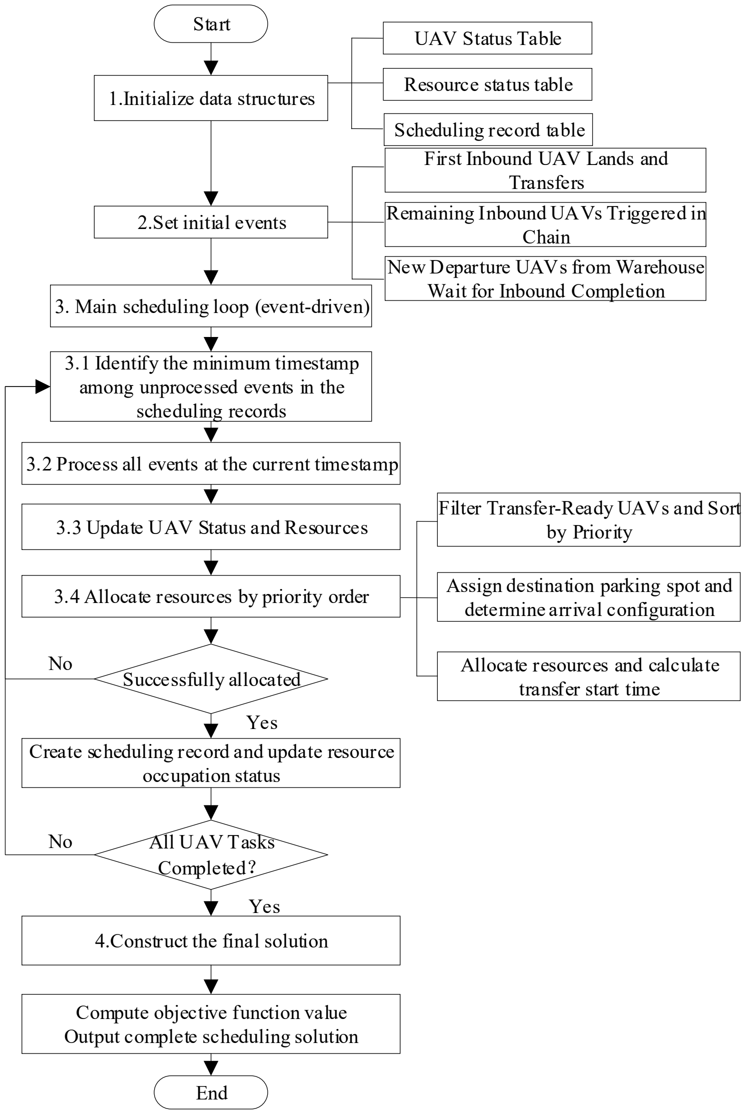


Figure 7. Flowchart of the EDCG initial solution generation strategy.

Step 1. Initialize current time $t = 0$, UAV status array *status*, UAV current phase array *stage*, configuration-space state $\Omega \leftarrow \Omega_0$, recovery completion flag $rec \leftarrow false$, and scheduling record $sched \leftarrow \varnothing$. Determine the recovery sequence according to group priority and intra-group priority, and determine the recovery landing time T_{land} of recovery groups based on the same/different landing spot intervals and the interference matrix. Create the recovery task for the first UAV; initialize the newly sortied UAV from the hangar: set status to blocked, configuration to wings folded, occupying the initial hangar parking spot. If initial event creation is completed, proceed to Step 2.

Step 2. While there exist UAV that have not completed scheduling, retrieve the next event time t and all events at that time from the scheduling record, and proceed to Step 3; otherwise, proceed to Step 8.

Step 3. Traverse each event η in *events*. If η is a start event, release the starting parking spot, propagate along the constraint propagation network G_{cons} and update UAV status, occupy tractor and elevator resources, and update UAV configuration; if the current phase is the recovery phase, trigger the landing of the next UAV. If η is an end event, release tractor resources, occupy the target parking spot, propagate along G_{cons} and update UAV status; if all UAV have landed and $rec \leftarrow false$, then set $rec \leftarrow true$, and update *status*; if the target parking spot is a support spot, create a support operation task. Update the current phase of the UAV. After all events at the current time are processed, proceed to Step 4.

Step 4. Filter the set of UAV $A_{avail} = \{i | status_i = available\}$ that are in schedulable status. Sort A_{avail} according to priority. If $A_{avail} \neq \varnothing$, proceed to Step 5; otherwise, proceed to Step 2.

Step 5. Traverse each UAV *i* in A_{avail} according to priority. Invoke the parking spot selection function to determine the target parking spot *S*, the post-arrival configuration c', and the transfer time t_z, with the calculation formula given as Equation (21):

$$(S, c') = \arg\min_{s \in S_i, c \in C_i} \{t_{ij}(s) | \Omega(s, c) = available\} \tag{21}$$

where S_i denotes the set of available parking spots for UAV *i* in the current phase, C_i denotes the set of available configurations, and $\Omega(s,c)$ represents the spatial state when parking spot *s* is occupied with configuration *c*. If the allocation is successful, invoke the resource allocation function to determine the start time t_s, completion time t_e, and the assigned tractor and elevator; append the decision record to *sched*, and update the resource states and configuration-space state Ω. If resource allocation fails, mark the UAV as blocked and proceed to traverse the next UAV. After all schedulable UAV in the current round have been traversed, proceed to Step 6.

Step 6. Check whether any new events exist in the current scheduling record. If so, proceed to Step 2; otherwise, proceed to Step 7.

Step 7. Advance the current time to the next earliest event occurrence time, and proceed to Step 2.

Step 8. The scheduling record *sched* constructs a complete solution S_0; compute the comprehensive objective function value, and output a feasible initial scheduling scheme.

The above greedy strategy generates an initial solution that is feasible but not necessarily optimal, which will subsequently be iteratively optimized through the destruction and repair operators of the ALNS algorithm.

6.2. Operator Design

6.2.1. Destruction Operator Design

The destruction operator set includes the following:

(1) Random destruction: From the scheduling scheme, the fixed recovery sequence and the support procedures tied to transfer operations are excluded, and a random number of tasks are removed from the remaining free procedures.

(2) Phase-related destruction: After a certain procedure of an UAV is selected using the random destruction operator, that procedure and all subsequent transfer procedures of that UAV are removed as a whole. Since the starting parking spot of a subsequent procedure of the same UAV is the target parking spot of its preceding procedure, and its start time is constrained by the completion time of the preceding procedure, removing only an intermediate procedure while retaining the subsequent ones would result in logical discontinuities and infeasible solutions. The block-wise removal approach provides room for re-planning the entire path from the breakpoint during the repair phase, which helps maintain the structural integrity of the solution and facilitates escaping from local optima.

(3) Maximum waiting-time destruction: Identify the procedure with the longest waiting time between adjacent transfer procedures in the scheduling scheme and remove it.

In platform operations, prolonged waiting typically arises from competition and imbalanced allocation of critical resources such as parking spots, tractors, or elevators, indicating that there is room for optimization in resource assignment or temporal arrangement. After removing this procedure and reallocating resources, unnecessary waiting can be compressed, making the overall schedule more compact while also improving resource load balancing.

The above three destruction operators are illustrated in Figure 8.

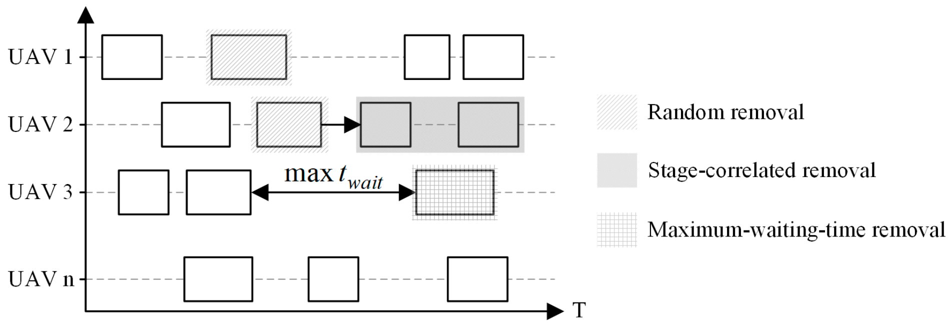


Figure 8. Schematic diagram of partial destruction operators.

(4) High-resource-occupancy destruction: The total usage time τ_t of each tractor is statistically calculated, and the selection probability is determined according to Equation (22), where tractors with longer usage times are assigned higher probabilities of being selected for destruction.

$$p_t = \frac{\tau_t}{\sum_{t' \in \mathrm{T}} \tau_{t'}}, \quad \forall t \in \Gamma \tag{22}$$

where Γ denotes the set of tractors. After a tractor t^* is selected, the random destruction operator is employed to remove some of the transfer procedures assigned to it, so that during the repair phase, part of the tasks of the overloaded tractor can be reassigned to idle tractors, thereby achieving load balancing.

6.2.2. Repair Operator Design

The repair operator set includes the following:

(1) Random repair: For each procedure to be repaired, a set of allocation schemes is randomly selected from the feasible combinations of parking spots, tractors, and time slots to generate a feasible solution.

(2) Greedy repair: According to the traversal order of procedures, for each procedure to be repaired, all feasible allocation schemes are evaluated, and the scheme that minimizes the start time of that procedure is selected for immediate execution, with resource states updated accordingly.

(3) Regret-value repair: Evaluate the marginal cost Δ_{jk} of each procedure j to be repaired at all feasible insertion positions, and after sorting, compute the k-regret value $R_j^{(k)} = \Delta_j^{(k)} - \Delta_j^{(1)}$. In each iteration, the procedure with the largest regret value is inserted first, and the scheduling scheme is updated; this process is repeated until all procedures are repaired.

(4) Critical-path-first repair: Construct a procedure dependency graph, and calculate the earliest start time ES_j and the latest start time LS_j of each procedure through forward recursion and backward recursion, respectively; then compute the slack time $\Delta t = LS_j - ES_j$, and the procedures with zero slack time constitute the critical path.

The procedures on the critical path are filtered out and repaired first, so as to avoid delays on the critical path from affecting the total transfer time, while the remaining procedures are subsequently allocated according to the greedy strategy.

6.3. Adaptive Weights

A combination of the roulette wheel selection strategy and the adaptive weight mechanism is adopted [30]. In the initial stage, equal weights are assigned to each operator, which are then dynamically adjusted according to the degree of improvement in the objective function. Operators with superior performance are assigned higher selection probabilities, guiding the algorithm to explore search spaces with high potential.

6.4. Acceptance Criterion

To avoid the algorithm from becoming trapped in local optima, the Metropolis acceptance criterion from simulated annealing is adopted [31]: a new solution is directly accepted if it is better than the current one; otherwise, it is accepted with probability $\exp\left(\frac{-\Delta f}{T}\right)$, where Δf is the increment in the objective function and T is the current temperature. The temperature is attenuated according to $T=\alpha T$, with α being the cooling coefficient. The search is terminated when the number of iterations reaches the upper limit or the temperature drops to the preset lower bound. The initial temperature is calculated as Equation (23):

$$T_0=\beta\frac{f\left(s_0\right)}{\ln\left(0.7\right)} \tag{23}$$

where β is the relative deviation coefficient, and $f\left(s_0\right)$ is the objective function value of the initial solution.

7. Numerical Simulation

To validate the effectiveness of the EDGC-ALNS algorithm in solving the platform area staging transition and collaborative scheduling problem of recovered UAV fleets, multiple simulation test cases with different scales and different delivery group configurations are set up, and the performance of the EDGC-ALNS algorithm is compared with that of the traditional ALNS algorithm and the genetic algorithm. The traditional ALNS algorithm adopts random destruction and phase-related destruction strategies, combined with random repair and greedy repair, and employs the simulated annealing Metropolis criterion as the acceptance criterion. The genetic algorithm adopts exchange, crossover, and mutation strategies, with a population size of 50, a mutation rate of 0.1, and a crossover rate of 0.8. The maximum number of iterations for all three algorithms is set to 1000, and a comparison is drawn based on the experimental results, with conclusions subsequently derived.

7.1. Simulation Scenario Setup

Matlab 2024B is used as the experimental platform, running on a computer with 16 GB memory and a 2.2 GHz processor. The simulation object is an offshore wind power substation. The platform is equipped with 37 temporary parking spots, 26 one-stop support parking spots, and 21 pre-departure preparation parking spots, while the hangar is equipped with 30 parking spots. The entire substation is equipped with 3 elevators (each capable of accommodating at most 2 UAVs simultaneously), 4 tractors in the platform area and 4 in the hangar, as well as 1 fixed-wing UAV landing spot and 3 multi-rotor UAV landing spots. The platform and hangar layouts is referred in Figure 4. Based on the path library constructed in Section 5.1, this experiment generates a total of 2159 platform-area transfer paths and 62 hangar transfer paths for the above parking spot layout, and the number of paths and their distribution among parking spots are shown in Figure 9.

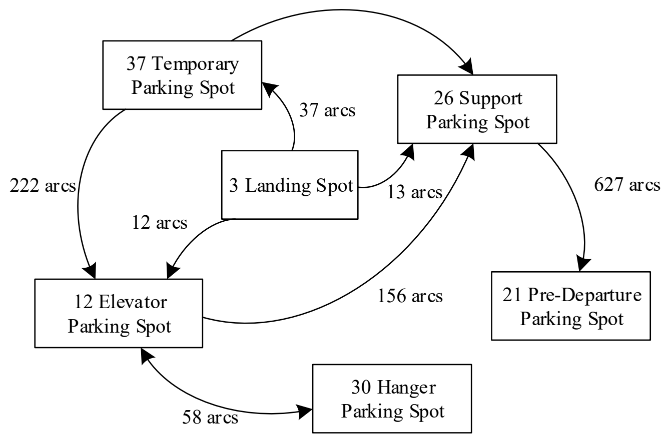


Figure 9. Schematic diagram of the simulation experiment dataset composition.

The experimental scenario is set as the platform-area staging transition and collaborative scheduling of the recovered multi-UAV fleet for re-sortie in confined space: among the recovered fleet on that day, some UAVs are required to be positioned on the platform for standby after support completion, while others with faults or no missions need to be transferred to the hangar for stowage; simultaneously, some UAVs originally parked in the hangar are required to be moved out, complete support operations, and then be positioned on the platform. The initial parking spot, target state, and support requirements of each UAV are predetermined.

The simulation parameters are set as follows: tying/untying time $tx = 30$ s, UAV configuration transformation time $tw = 20$ s, elevator single-trip transfer time $te = 70$ s, continuous recovery interval at the same landing spot $\delta_{es} = 120$ s, interval at different landing spots $\delta_{ed} = 20$ s, collision detection step size $\Delta t = 1$ s. The maximum towing speed of tractors under load is $v_{load} = 1$ m/s, the maximum speed under no load is $v_{empty} = 3$ m/s, the maximum acceleration is $a = 1$ m/s^2. The platform and hangar are each equipped with 4 tractors, and the entire substation is equipped with 3 elevators. there is 1 fixed-wing landing spot and 3 multi-rotor UAV landing spots.

The operational formation settings are shown in Table 5. Four groups undertake offshore inspection, emergency patrol, material delivery A, and material delivery B, respectively. The recovery priority and launch priority of each group are consistent. The small-scale formation set includes the offshore inspection and material delivery A groups, the medium-scale set adds the emergency patrol group, and the large-scale case incorporates all four groups.

Table 5. Operational group settings.

Formation Designation	UAV Type Composition	Payload Mounting Mode	Recovery Priority	Launch Priority	Servicing Duration/s
Offshore inspection	Large fixed-wing UAV	-	1	1	900
Emergency patrol	Medium multi-rotor UAV	-	2	2	780
Material delivery A	Foldable-wing UAV	Wingtip-mounted payload	3	3	1200
Material delivery B	Foldable-wing UAV	Belly-mounted payload	4	4	1200

The small-scale formation set includes the offshore inspection and material delivery A groups; the medium-scale set adds the emergency patrol group; and the large-scale case incorporates all four groups. The number of UAVs within each group is proportionally allocated according to the total scale. The number of faulty UAVs is set to 15–25% of the recovered fleet size, and the group composition of newly launched UAVs is kept symmetric to that of faulty UAVs, so as to ensure consistency between the recovered and the launched group sizes.

As shown in Table 6, to validate the effectiveness of the EDGC-ALNS algorithm, six simulation test cases are set up. Among them, Cases 1–3 are baseline scenarios, corresponding to small, medium, and large scales, respectively; Cases 4–6 add special settings based on the baseline scenarios to examine the alternate configuration layout strategy, the two-phase coupling mechanism, and the algorithm robustness under resource-constrained conditions, respectively. Each case is independently run 30 times, and the mean and standard deviation of the objective function values are taken as the evaluation metrics.

Table 6. Simulation experiment test cases.

Index	Recovery Formation Size	Faulty/No-Mission UAVs	New Launch UAVs	Launch Formation Size	Formation Quantity	Special Configuration
Case 1	6	1	1	6	2	Standard 2-UAV formation
Case 2	12	3	3	12	3	Standard 3-UAV formation
Case 3	20	5	5	20	4	Standard 4-UAV formation
Case 4	12	3	3	12	3	Available service parking spots reduced by 30%
Case 5	12	3	3	12	3	Recovery and launch formation priorities are reversed
Case 6	20	5	5	20	4	Number of platform-area tractors reduced by half

7.2. Hyperparameter Experiment

The solution performance of the ALNS algorithm is significantly influenced by the settings of its hyperparameters. To determine the optimal parameter combination for the problem addressed in this paper, an orthogonal experimental design is adopted to conduct a systematic analysis of the key hyperparameters.

Five hyperparameters that have a notable impact on algorithm performance are selected as experimental factors: population size Np, operator weight learning factor s, destruction operator removal ratio P_{m1}, simulated annealing cooling coefficient P_{m2}, and the maximum consecutive non-improving iteration trigger threshold K. Each factor is assigned four levels, covering a reasonable range of parameter values, as shown in Table 7. An $L_{16}(4^5)$ orthogonal array is employed to generate 16 parameter combinations. For each combination, 20 independent runs are conducted under a 12-UAV sortie task (corresponding to the scale of Case 2 in Table 6), with the termination condition uniformly set as the scheduling evaluation count reaching 20,000 times. The mean value of the comprehensive objective function C_{max} obtained from each run is taken as the evaluation metric for that parameter combination.

Table 7. Orthogonal table of algorithm parameters.

	Np	s	P_{m1}	P_{m2}	K
1	10	0.25	0.05	0.05	3
2	10	0.50	0.30	0.30	5
3	10	0.75	0.50	0.50	10
4	10	1.00	1.00	1.00	20
5	30	0.25	0.30	0.50	20
6	30	0.50	0.05	1.00	10
7	30	0.75	1.00	0.05	5
8	30	1.00	0.50	0.30	3
9	50	0.25	0.50	1.00	5
10	50	0.50	1.00	0.50	3
11	50	0.75	0.05	0.30	20
12	50	1.00	0.30	0.05	10
13	100	0.25	1.00	0.30	10
14	100	0.50	0.50	0.05	20
15	100	0.75	0.30	1.00	3
16	100	1.00	0.05	0.50	5

The experimental results for the 16 parameter combinations are presented in Table 8. All $C_{\max}$ values listed are normalized composite objective function values, where a smaller value indicates superior scheduling performance under the corresponding parameter configuration. To systematically investigate the influence of each factor on algorithmic performance and their relative importance, a range analysis is conducted on the orthogonal experimental data. For each factor j, the sum of $C_{\max}$ over all experiments at levels 1 through 4 is computed, denoted as O_{1j}, O_{2j}, O_{3j} and O_{4j}, respectively, from which the range R_j is derived. A larger R_j signifies a more pronounced effect of variations in that factor on algorithmic performance, thereby enabling a quantitative ranking of factor importance.

Table 8. Orthogonal experiment parameter combinations and results.

	Np	s	P_{m1}	P_{m2}	K	$C_{\max}$
1	10	0.25	0.05	0.05	3	0.7789
2	10	0.50	0.30	0.30	5	0.7725
3	10	0.75	0.50	0.50	10	0.7598
4	10	1.00	1.00	1.00	20	0.7462
5	30	0.25	0.30	0.50	20	0.7624
6	30	0.50	0.05	1.00	10	0.7796
7	30	0.75	1.00	0.05	5	0.7556
8	30	1.00	0.50	0.30	3	0.7647
9	50	0.25	0.50	1.00	5	0.7664
10	50	0.50	1.00	0.50	3	0.7493
11	50	0.75	0.05	0.30	20	0.7701
12	50	1.00	0.30	0.05	10	0.7804
13	100	0.25	1.00	0.30	10	0.7605
14	100	0.50	0.50	0.05	20	0.7538
15	100	0.75	0.30	1.00	3	0.7741
16	100	1.00	0.05	0.50	5	0.7856
O_{1j}	3.0574	3.0682	3.1142	3.0687	3.0670	—
O_{2j}	3.0623	3.0552	3.0894	3.0737	3.0801	—
O_{3j}	3.0662	3.0496	3.0447	3.0571	3.0803	—
O_{4j}	3.0740	3.0770	3.0117	3.0522	3.0325	—

https://doi.org/10.3390/machines14101107

R_j	0.0066	0.0095	0.0213	0.0118	0.0145	—
Primary—Secondary factor	$P_{m1} > K > s > P_{m2} > Np$					
Optimal scheme	$Np = 50$, $s = 0.75$, $P_{m1} = 1.0$, $P_{m2} = 0.5$, $K = 20$					

From the range data presented in Table 8, the factors can be ordered by their influence on scheduling effectiveness as follows: destruction removal ratio P_{m1} > restart threshold K > learning factor s > cooling coefficient P_{m2} > population size Np. Among these, the destruction removal ratio exerts the most dominant influence, with its range value (0.0213) being approximately 3.2 times that of the population size (0.0066). This indicates that the perturbation intensity of neighborhood search is the critical determinant of algorithm convergence quality—an excessively small removal ratio yields insufficient neighborhood variation, rendering the algorithm prone to entrapment in local optima, whereas an overly large removal ratio may excessively disrupt solution structural integrity, thereby increasing the difficulty of repair. Consequently, the removal ratio must be judiciously calibrated to strike a balance between exploration and exploitation. Notably, the population size exerts the least influence among all factors, suggesting that the algorithm is insensitive to population cardinality and maintains satisfactory performance even with a relatively small population. This characteristic is advantageous for reducing computational overhead, enabling EDGC-ALNS to achieve competitive convergence within a limited number of evaluations.

For each factor, the level corresponding to the minimum O_{ij} among its four levels is selected as the optimal level, representing the level that yields the best average scheduling performance. Synthesizing the optimal levels across all factors, the optimal parameter combination is determined as: $Np = 50$, $s = 0.75$, $P_{m1} = 1.0$, $P_{m2} = 0.5$, $K = 20$. Under this configuration, the algorithm achieves a well-coordinated balance among neighborhood search intensity, weight update rate, annealing cooling rhythm, and restart triggering conditions. All subsequent algorithm comparison experiments uniformly adopt this parameter configuration to ensure fairness and reproducibility of the comparative results.

7.3. Pareto Sensitivity Analysis of the Objective Weights

The overall objective function in this paper is constructed by linearly weighting three normalized sub-objectives. Because the three sub-objectives differ significantly in physical dimension and numerical magnitude, direct weighting would allow the sub-objective with the largest magnitude to dominate the optimization direction. Specifically, the total transfer time F_1 is the latest completion time of all transfer operations, the number of configuration changes F_2 is the cumulative number of configuration switches of the fleet across all transfer stages, and the load variance among transfer crews F_3 is measured as the variance of tractor working time in minutes. The three sub-objectives are approximately on the order of 10^3, 10^1, and 10^0, respectively. If directly weighted, F_1 would dominate the objective function due to its absolute magnitude, severely weakening the optimization effects of configuration changes and load variance.

To address this, each sub-objective is normalized to a unified dimensionless interval before weighted aggregation. The normalized sub-objective is defined as Equation (24):

$$\tilde{F}_i = \frac{F_i}{N_i}, i = 1, 2, 3 \tag{24}$$

where N_i is the normalization benchmark of the i-th sub-objective. The benchmark is selected such that the normalized values of the three sub-objectives fall within a comparable range under typical scheduling scenarios, ensuring that the weight coefficients reflect the decision-maker's relative preferences rather than being passively dominated by numerical magnitudes. Based on the preliminary statistical results of the 21-aircraft case and the theoretical lower bounds, we set N_1 = 5000, N_2 = 10 and N_3 = 1. On this basis, the overall objective function is shown as Equation (5).

The weight coefficients λ_i directly determine the optimization trajectory. To avoid subjective weighting, a Pareto sensitivity experiment is designed, in which the weight combination is systematically varied and the responses of the three sub-objectives and the overall objective are observed, so that the final weights can be determined from the experimental results. The experiment is conducted on the 21-aircraft case with fixed algorithm parameters and random seeds, varying only the weight combination. The weight interval settings are related to the variation amplitudes of the three sub-objectives: the total transfer time F_1 is on the order of thousands, but the percentage improvement in its value during iteration is relatively small, so its marginal impact on the overall objective function after normalization is relatively limited; the number of configuration changes F_2 and the load variance F_3 are on the order of single digits to tens, and their iterative improvements are relatively significant relative to their total values, so their impact on the overall objective function after normalization is more pronounced. If comparable weight intervals were adopted for the three sub-objectives, the slight improvements in F_2 and F_3 would be amplified after normalization and dominate the descent direction of the overall objective function, while the continuous optimization of F_1 would be difficult to reflect in the overall objective function.

Therefore, the weight combinations are generated as follows: λ_1 ranges from 0.70 to 0.98, λ_2 ranges from 0.01 to 0.20, and $\lambda_3 = 1 - \lambda_1 - \lambda_2$ with $\lambda_3 \geq 0.01$. For each weight combination, the algorithm is run independently 30 times, and take the average value as the objective value under that combination. Representative results are shown in Table 9.

Table 9. Sub-objective and overall objective values under different weight combinations.

No.	λ_1	λ_2	λ_3	F_1	F_2	F_3	$\tilde{F}_1$	$\tilde{F}_2$	$\tilde{F}_3$	**Overall** F
1	0.68	0.2	0.12	5538.6	9.4	0.76	1.10772	0.94	0.76	1.039
2	0.72	0.18	0.10	5462.1	9.5	0.77	1.09242	0.95	0.77	1.037
3	0.76	0.15	0.09	5391.5	9.5	0.8	1.0783	0.95	0.8	1.037
4	0.80	0.12	0.08	5348.3	9.7	0.82	1.06966	0.97	0.82	1.041
5	0.84	0.10	0.06	5303.7	9.7	0.82	1.06074	0.97	0.82	1.038
6	0.86	0.09	0.05	5256.4	9.9	0.83	1.05128	0.99	0.83	1.038
7	**0.9**	**0.07**	**0.03**	**5198.6**	**10**	**0.85**	**1.03972**	**1**	**0.85**	**1.031**
8	0.92	0.06	0.02	5192.4	10.2	0.89	1.03848	1.02	0.89	1.034
9	0.94	0.05	0.01	5187.3	10.8	0.94	1.03746	1.08	0.94	1.039
10	0.96	0.03	0.01	5181.9	11.3	1	1.03638	1.13	1	1.039
11	0.98	0.01	0.01	5176.8	12	1.07	1.03536	1.2	1.07	1.037

As shown in Table 9, as the total transfer time weight λ_1 increases from 0.68 to 0.98, the three sub-objectives exhibit a clear trade-off relationship. The total transfer time F_1 decreases from 5538.6 s to 5176.8 s, a reduction of about 6.5%; the number of configuration changes F_2 increases from 9.4 to 12.0, rising by about 27.7%; and the load variance F_3 increases from 0.76 to 1.07, rising by about 40.8%. Increasing λ_1 can effectively compress the task completion time, but leads to more frequent configuration switches and worse load balancing. The overall objective value F exhibits a U-shaped trend as λ_1 increases: when λ_1 increases from 0.70 to 0.90, F decreases from 1.0394 to 1.0312; when λ_1 further increases to 0.98, F rebounds to 1.0374. When λ_1 is small, the relatively large F_1 drags down the overall objective; once λ_1 exceeds 0.90, the marginal improvement in F_1

becomes limited, while F_2 and F_3 continue to rise, and their weighted contributions gradually offset the improvement in F_1.

The combination λ_1 = 0.90, λ_2 = 0.07, λ_3 = 0.03 yields an overall objective value *F* = 1.0312, the lowest among the 11 weight combinations. Under this combination, F_1 = 5198.6 s, close to the single-objective optimum of 5176.8 s; F_2 = 10.0, within a preferable range; and F_3 = 0.85, also within a reasonable range. Therefore, the combination λ_1 = 0.90, λ_2 = 0.07, λ_3 = 0.03 is selected for the overall objective function, which ensures the primary optimization role of total transfer time while preserving effective optimization space for configuration changes and load variance.

7.4. Computational Complexity Analysis

To theoretically evaluate the solution efficiency of the EDGC-ALNS algorithm under problems of different scales, this section analyzes the time complexity of each of its modules.

Let the problem scale be defined as the total number of UAV participating in scheduling *n*, the total number of parking spots *m*, the number of unmanned tractors *r*, and the number of elevators *e*. In the simulation scenario of this paper, *m* is approximately linearly related to *n* (the total number of available parking spots increases proportionally with the scale of UAV), while *r* and *e* are constants.

The initial solution generation part of the EDCG algorithm adopts an event-driven mechanism to advance the scheduling process according to events. The processing of each transfer procedure involves the following operations: selecting an UAV from the current set of schedulable UAV according to priority ($O(n)$), traversing available parking spots to evaluate compatibility ($O(m)$), and matching available tractor and elevator resources ($O(r+e)$). The number of transfer procedures experienced by each UAV is a constant *p* (typically $p \le 4$), and the total number of procedures is $N = p \cdot n = O(n)$. Therefore, the time complexity of the EDCG initial solution generation is shown as Equation (25):

$$T_{EDGC} = O\left(N \cdot (n+m+r+e)\right) = O\left(n \cdot (n+n+1)\right) = O\left(n^2\right) \tag{25}$$

The ALNS main loop executes a total of *I* iterations. In a single iteration, the destruction operator removes *q* procedures from the current solution ($q = O(n)$), and the repair operator reinserts the removed procedures. Each insertion requires traversing available parking spots ($O(m)$) and idle resources ($O(r+e)$) to evaluate all feasible allocations. The repair phase processes the *q* procedures to be repaired one by one, and the complexity of a single repair is $O(m+r+e) = O(n)$. Thus, the complexity of a single iteration is $O\left(n^2\right)$, and the total complexity of the main loop is shown as Equation (26):

$$T_{ALNS} = O\left(I \cdot n^2\right) \tag{26}$$

The candidate solution generated in each iteration requires evaluation by invoking the comprehensive objective function. The objective function comprises three terms: total transfer time, number of configuration changes, and load variance. The evaluation entails traversing all procedures to count time and configuration changes ($O(n)$) and aggregating the load of each tractor ($O(n)$), resulting in an overall evaluation complexity of $O(n)$. Based on the above analysis, the overall time complexity of the EDGC-ALNS algorithm is shown as Equation (27):

$$T_{EDGC-ALNS} = O\left(n^2\right) + O\left(I \cdot n^2\right) + O\left(I \cdot n\right) = O\left(I \cdot n^2\right) \tag{27}$$

Since the maximum number of evaluations I is a constant, the algorithm complexity primarily grows quadratically with the UAV scale n, which falls into the category of polynomial time complexity and is capable of solving medium-to-large-scale problems ($n \le 20$) within an acceptable time.

To validate the above theoretical analysis, the average runtime of the EDGC-ALNS algorithm is recorded under three standard scenarios of different scales (Case 1–Case 3), and the results are shown in Table 10. As the problem scale increases from 6 to 20 UAV, the algorithm runtime exhibits a nonlinear growth trend: the runtime of Case 2 (12 UAV) is approximately 2.8 times that of Case 1, and that of Case 3 (20 UAV) is approximately 8.7 times that of Case 1. This growth trend is generally consistent with the theoretical quadratic complexity $O\left(n^2\right)$. The actual runtime is lower than the theoretical worst-case estimate, which is mainly attributed to the following factors: first, the EDCG initial solution generation strategy substantially compresses the search space of feasible solutions; second, the fixed numbers of tractors and elevators limit the exponential growth of resource allocation branches; and third, the problem scales have not yet reached the asymptotic dominant region, where constant terms and lower-order terms still account for a certain proportion. In summary, the time complexity of the EDGC-ALNS algorithm is controllable in practical solving and can meet the engineering application requirements for medium-to-large-scale problems.

Table 10. Algorithm runtime under different scales.

Test Case	Total Number of UAV	Average Runtime/s
Case 1	6	1.16
Case 2	12	3.24
Case 3	20	10.1

7.5. Algorithm Performance Comparison

7.5.1. Comparison of Solution Quality and Real-Time Performance

To comprehensively evaluate the solution performance of the EDGC-ALNS algorithm, a comparative analysis is conducted from three dimensions: solution quality, stability, and computational efficiency. The comparison algorithms include the traditional ALNS and the genetic algorithm (GA). All algorithms are run under the same experimental environment, and each case is independently run 30 times. The mean objective function value (Avg), standard deviation (Std), and average runtime (Time) are taken as the evaluation metrics, and the results are shown in Table 11.

Table 11. Comparison of algorithm solution results for test cases of different scales.

Experiment Index	EDGC-ALNS			ALNS			GA		
	Avg	Std	Time	Avg	Std	Time	Avg	Std	Time
Case1	0.52	0.04	1.16	0.57	0.07	1.38	0.62	0.09	0.68
Case2	0.74	0.07	3.24	0.93	0.09	4.05	1.04	0.19	2.15
Case3	1.07	0.14	10.1	1.23	0.19	12.73	1.44	0.28	6.72
Case4	0.78	0.08	7.35	0.94	0.14	8.97	1.11	0.31	4.9
Case5	0.86	0.08	9.5	1.01	0.15	11.88	1.15	0.33	6.28
Case6	1.34	0.17	16.8	1.53	0.37	20.58	1.85	0.51	10.9

(1) Solution quality comparison

The objective function values in Table 11 have been normalized, with smaller values indicating better comprehensive scheduling performance. Among all test cases, EDGC-

ALNS achieves the smallest mean objective function value and the lowest standard deviation among the three algorithms, demonstrating its superiority over the comparison algorithms in both solution quality and stability.

Cases 1 to 3 are standard scenarios with increasing scales. EDGC-ALNS achieves improvements of 7.0–7.9% over the traditional ALNS and 14.4–22.1% over the genetic algorithm. As the problem scale increases, the standard deviation of the genetic algorithm increases from 0.15 to 0.44, representing the largest increase, which reflects its insufficient search stability for large-scale problems; the standard deviation of EDGC-ALNS only increases from 0.05 to 0.14, demonstrating the best fluctuation control capability. In terms of runtime, EDGC-ALNS is slightly higher than the genetic algorithm but lower than the traditional ALNS, and this time cost is within an acceptable range.

Case 4 is a scenario where the number of support parking spots is reduced by 30%. The objective function value of EDGC-ALNS increases by 5.4% compared with Case 2, while those of the traditional ALNS and the genetic algorithm increase by 9.3% and 8.8%, respectively. Under parking spot scarcity, EDGC-ALNS adopts the alternate wing-spreading layout strategy in limited space, allowing UAVs with wingtip-mounted payload to remain in place after payload mounting at the support spots. At the cost of a small increase in the number of configuration changes, additional transfer scheduling is avoided, effectively controlling the increase in the overall objective function value.

In Case 5, where the recovery and launch group priorities are reversed, the objective function values of the three algorithms increase by 16.2%, 17.4%, and 12.7%, respectively, compared with Case 2, representing a relatively significant increase. UAVs with high recovery priority but low launch priority must occupy temporary parking spots immediately after landing. Due to the path blocking relationships among parking spots, the outbound transfer and support scheduling of subsequent UAVs with high launch priority are inevitably delayed, slowing down the overall scheduling pace. The standard deviation of EDGC-ALNS in this scenario is 0.08, comparable to that in Case 4, indicating that the initial solution strategy can still generate feasible schemes under temporally coupled scenarios.

In Case 6, where the number of platform-area tractors is halved, the objective function values of the three algorithms increase by 25.2%, 20.5%, and 26.7%, respectively, compared with Case 3, representing the largest increase among all test cases. The shortage of tractor resources results in a large number of UAVs waiting for transfer, significantly prolonging the overall transfer time. The increased conflicts in tractor allocation lead to greater fluctuation in scheduling schemes, and the standard deviations of all algorithms rise markedly. Nevertheless, EDGC-ALNS, through the high-resource-occupancy destruction operator, reassigns part of the tasks of overloaded tractors, maintaining relatively optimal scheduling quality under resource-constrained conditions.

(2) Solution quality distribution characteristics

To further examine the stability performance of the three algorithms under different scales, the results of 30 independent runs for Cases 1 to 3 are selected to generate boxplots, as shown in Figure 10.

From the overall distribution, the box length of EDGC-ALNS at each scale is significantly shorter than those of the traditional ALNS and the genetic algorithm, and its interquartile ranges are compact, indicating smaller fluctuations in solution quality. In terms of outliers, EDGC-ALNS and the traditional ALNS exhibit no outliers, while the traditional ALNS shows a few sporadic outliers, and the genetic algorithm presents a relatively large number of outliers, reflecting that the genetic algorithm's search process for large-scale problems is unstable and prone to converging to local optima of varying quality.

From the perspective of scale adaptability, as the problem scale expands from Case 1 to Case 3, the box expansion of EDGC-ALNS is the smallest, indicating that it can still maintain high solution consistency in complex scheduling scenarios. Combined with the standard deviations of each algorithm in Table 11, EDGC-ALNS exhibits the smallest deviation of multiple runs from the mean at the same scale, verifying the improvement effect of the EDCG initial solution strategy and the adaptive weight mechanism on algorithm stability.

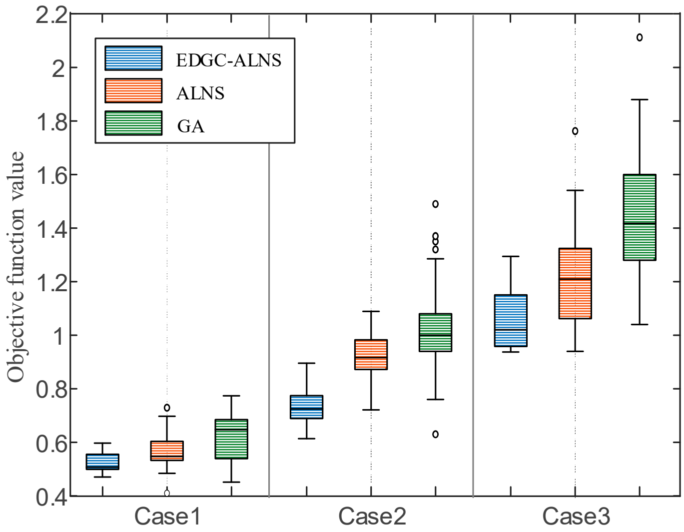


Figure 10. Boxplot of algorithm comparison.

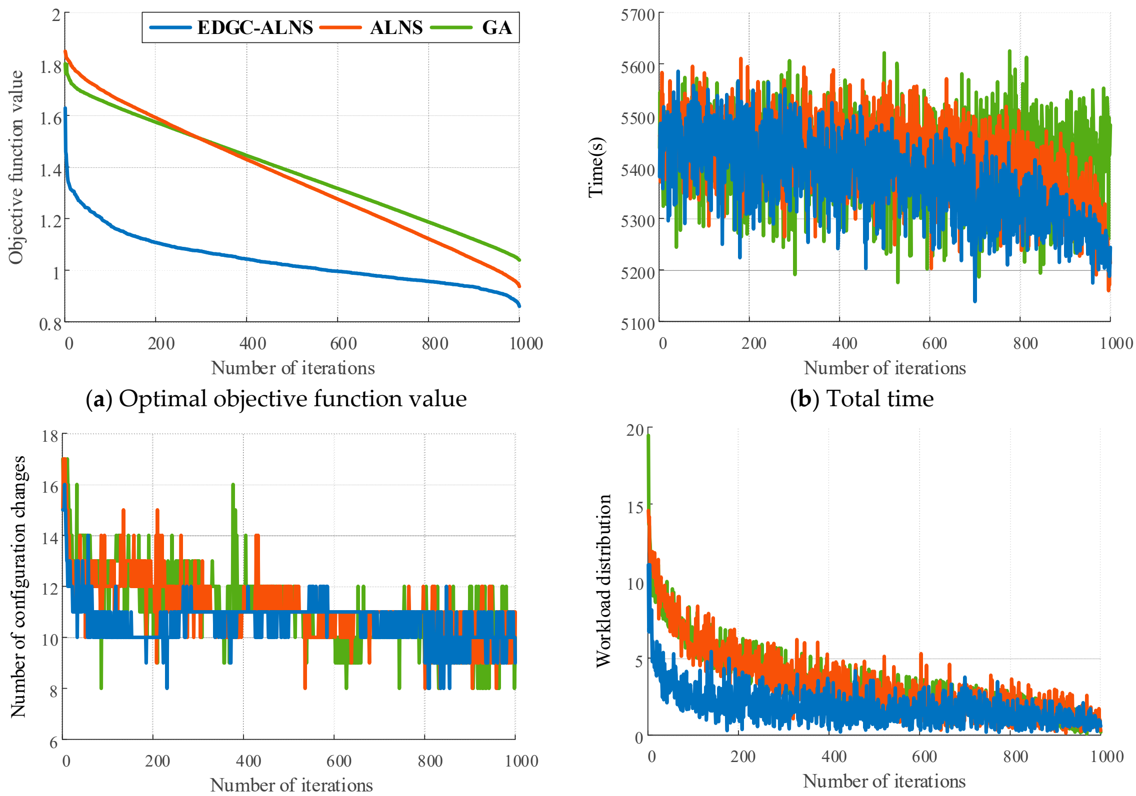
(3) Convergence performance analysis

To further examine the convergence behavior and optimization efficiency of each algorithm, Case 3 (20-UAV scale) is selected as a representative test case, and the variation trajectories of the objective function values of the three algorithms over 1000 iterations are recorded, with the results shown in Figure 11. Figure 11a presents the convergence curves of the overall objective function value, with the vertical axis representing the normalized objective function value; Figure 11b–d respectively present the convergence processes of the three sub-objectives (total transfer time, number of configuration changes, and load variance among transfer crews), with the vertical axes representing the original sub-objective function values.

As shown in Figure 11a, the objective function value of EDGC-ALNS decreases rapidly in the early stage of iteration (approximately the first 200 iterations), with a convergence speed significantly faster than that of the traditional ALNS and GA. This is mainly attributed to the EDCG initial solution strategy, which provides a good starting point for the search. In the middle and late stages of iteration, EDGC-ALNS continues to decrease slowly and gradually stabilizes, eventually converging to the lowest value. The traditional ALNS and GA exhibit similar overall convergence trends, but the convergence depth of ALNS in the later stage is slightly better than that of GA, while EDGC-ALNS consistently maintains the lowest objective function value.

From the sub-objective convergence curves in Figure 11b–d, it can be observed that the three sub-objectives exhibit distinct convergence behaviors during the optimization process. The total transfer time (Figure 11b) has the highest weight, but its variation range in the early stage of iteration is limited; the total transfer times of the three algorithms are all approximately 5400 s, with a difference of only about 200 s among them, representing a relative change of less than 4%. After normalization, even when multiplied by the high weight of 0.9, its contribution to the overall objective function remains relatively limited, and thus it is not the primary source of the rapid initial decline. In contrast, the number of configuration changes (Figure 11c) and the load variance among transfer crews (Figure

11d), although assigned lower weights, both range between 0 and 20. Small improvements in the early stage of iteration can produce significant effects after normalization, becoming the dominant factors driving the rapid convergence of the overall objective function. After both tend to stabilize at approximately 500 iterations, the continued optimization of the total transfer time (Figure 11b) becomes the primary source for further improvement of the overall objective function. Among the three algorithms, EDGC-ALNS achieves the optimal convergence results across all sub-objectives, while GA exhibits larger fluctuations in total transfer time and iteration count, with insufficient convergence stability. Overall, the linear weighting strategy guides the algorithm to focus on the corresponding optimization directions according to sub-objective weights at different stages, and all sub-objectives converge to relatively good levels, without any single objective being excessively sacrificed.



(**a**) Optimal objective function value (**b**) Total time (**c**) Number of configuration changes (**d**) Variance of workload distribution

Figure 11. Convergence curves of the objective function.

In summary, EDGC-ALNS outperforms the comparison algorithms in terms of convergence speed, convergence depth, and stability. This benefits from the EDCG initial solution strategy providing a favorable starting point for the search, coupled with the adaptive weight mechanism and domain knowledge-driven operators, which effectively guide the search process toward high-quality solution regions.

(4) Real-time performance comparison

From Table 11, it can be seen that the runtime of EDGC-ALNS is slightly higher than that of GA but lower than that of the traditional ALNS. At the Case 3 (20-UAV) scale, the average runtime of EDGC-ALNS is 10.1 s, which is approximately 20.7% shorter than that of the traditional ALNS (12.73 s) and approximately 50.3% longer than that of GA (6.72 s). This time cost is within an acceptable range: on one hand, the 10-second-level solution time fully satisfies the scenario requirements for offline scheduling planning; on the other hand, the significant advantage of EDGC-ALNS in solution quality (improvement of 14.4–

22.1% over GA) is sufficient to compensate for its time overhead. In terms of growth trend, as the problem scale expands from Case 1 to Case 3, the runtime of EDGC-ALNS increases from 1.16 s to 10.1 s, representing an increase of approximately 8.7 times, which is generally consistent with the theoretical quadratic growth trend analyzed in Section 7.4.

Overall, EDGC-ALNS outperforms the comparison algorithms in both solution quality and stability. Although its runtime is slightly higher than that of GA, it is significantly lower than that of the traditional ALNS and remains within an acceptable range, demonstrating a good balance between solution accuracy and computational efficiency.

7.5.2. Statistical Significance Test

The comparison results in Section 7.5.1 indicate that EDGC-ALNS outperforms the traditional ALNS and GA in terms of the mean objective function value. However, comparison based solely on the mean and standard deviation is insufficient to statistically infer whether the differences among algorithms are significant, as the mean differences may arise from random fluctuations rather than substantive algorithmic improvements. To address this, the Wilcoxon rank-sum test is employed in this section to conduct statistical significance tests on the optimization results of each algorithm.

The Wilcoxon rank-sum test is a non-parametric testing method that does not require the data to follow a normal distribution and is suitable for comparing whether two independent samples originate from the same distribution. The testing procedure is as follows: first, the two groups of samples are pooled and sorted by numerical values to assign ranks; then, the rank sums of each group are calculated. If the rank sums of the two groups differ significantly, it indicates a significant difference between the two datasets. The test results are measured by the p-value, where a smaller p-value provides stronger evidence against the null hypothesis that "there is no difference between the two groups." In this paper, $p \le 0.05$ is adopted as the threshold for significant difference and $p \le 0.01$ for highly significant difference, with the two-sided test format applied.

With EDGC-ALNS as the baseline group, the tests are conducted against the traditional ALNS and GA, respectively. The tests are based on the objective function values from 30 independent runs under each test case, and the results are shown in Table 12.

Table 12. Wilcoxon rank-sum test results.

Comparison Group	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6
EDGC-ALNS vs. ALNS	0.008	0.023	0.046	0.006	0.009	0.038
EDGC-ALNS vs. GA	0.015	0.004	0.031	0.002	0.017	0.003

From Table 12, it can be observed that in all six test cases, the p-values between EDGC-ALNS and the traditional ALNS are all less than 0.05, among which the p-values for Case 1, Case 4, and Case 5 are below 0.01, indicating that the differences between the two are statistically significant. The comparison between EDGC-ALNS and GA shows consistent results: the p-values for all six cases are less than 0.05, and those for Case 2, Case 4, and Case 6 are below 0.01. The improvements of EDGC-ALNS over the traditional ALNS and GA are not attributable to random fluctuations but represent statistically significant advantages.

7.6. Large-Scale Formation Sortie Simulation

To intuitively present the solution results, Case 3 is taken as an example, and the optimal solution obtained by the EDGC-ALNS algorithm is decoded into an optimal scheduling scheme. This scenario involves 20 UAVs in both the recovery group and the launch group. The recovery group consists of four groups: offshore inspection (M1, M2), emergency patrol (Z1, Z2), material delivery A (J1–J8), and material delivery B (J9–J16). Among them, Z1, J2, J5, J6, and J12 are malfunctioning or mission-free UAVs and are required to be stored in the hangar, while Z3 and J17–J20 are newly launched UAVs from the hangar. The group priority, in descending order, is offshore inspection, emergency patrol, material delivery A, and material delivery B.

The Gantt chart of the scheduling results is shown in Figure 12. Inside each Gantt bar, the tractor or elevator used in the transfer process is labeled, where "TD" denotes platform-area tractor, "TH" denotes hangar tractor, and "E" denotes elevator; the numbers at the two ends of each Gantt bar represent the parking spot numbers before and after the transfer, respectively, and different colors distinguish different transfer/support tasks.

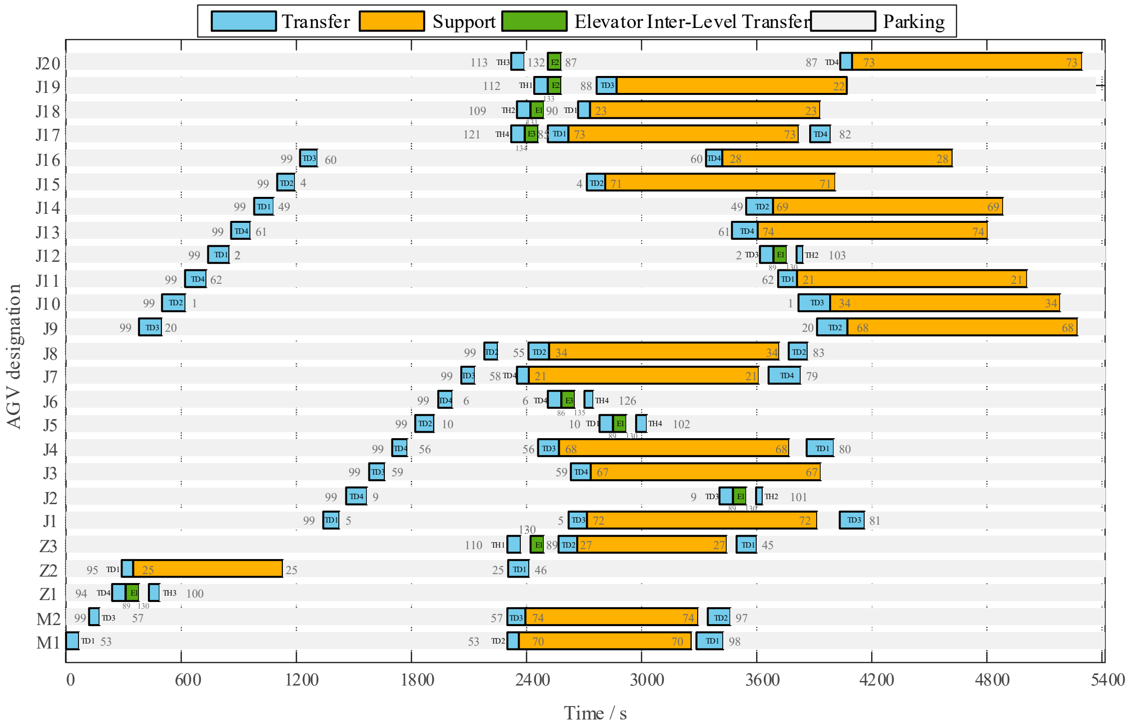


Figure 12. Gantt chart of the optimized results.

As shown in Figure 12, the recovery phase starts from time 0, and each group lands in sequence according to priority. Most UAVs are transferred to temporary parking spots after recovery, and subsequent transfers are not initiated until all recovery operations are completed, so as to avoid interfering with the landing paths. The faulty multi-rotor UAV Z1 is moved directly into the hangar and is transferred to the elevator immediately after landing.

During the support phase, both the material delivery B group (with belly-mounted payload) and the material delivery A group (with wingtip-mounted payload) have a support duration of 1200 s. The former can directly conduct payload mounting operations at the support spots, while the latter requires a configuration switch from wings folded to wings spread before support, resulting in a slightly longer actual occupancy time of the support spots. The alternate wing-spreading layout strategy effectively alleviates the additional demand for parking spot width caused by UAVs with wingtip-mounted payload, enabling support operations to proceed in an orderly manner within the limited space.

During the pre-departure preparation phase, each group, in order of priority, is transferred to the pre-departure parking spots after completing support. The offshore inspection and emergency patrol groups take their positions first, followed by the two material delivery groups in sequence. After the pre-departure parking spots become saturated, the remaining UAVs that have completed support remain in place at the support parking spots, maintaining their current configurations, and are transferred sequentially as space becomes available later.

To quantitatively evaluate the load balancing of tractor resources, Figure 13 presents the distribution of operating time of each tractor during the scheduling horizon. As shown in Figure 13, TD1–TD4 each undertake 12 to 14 transfer tasks, while TH1–TH4 each undertake 2 to 3 tasks, indicating a relatively balanced load distribution. This demonstrates that the algorithm, while adhering to the principle of nearest assignment, effectively balances the workload of each tractor through the high-resource-occupancy destruction operator, which reassigns tasks from overloaded tractors. The occupied periods of the elevators are concentrated in the hangar-entry window after recovery completion and the outbound transfer phase, and no conflicts arising from multiple UAVs contending for the same elevator simultaneously are observed among the three elevators.

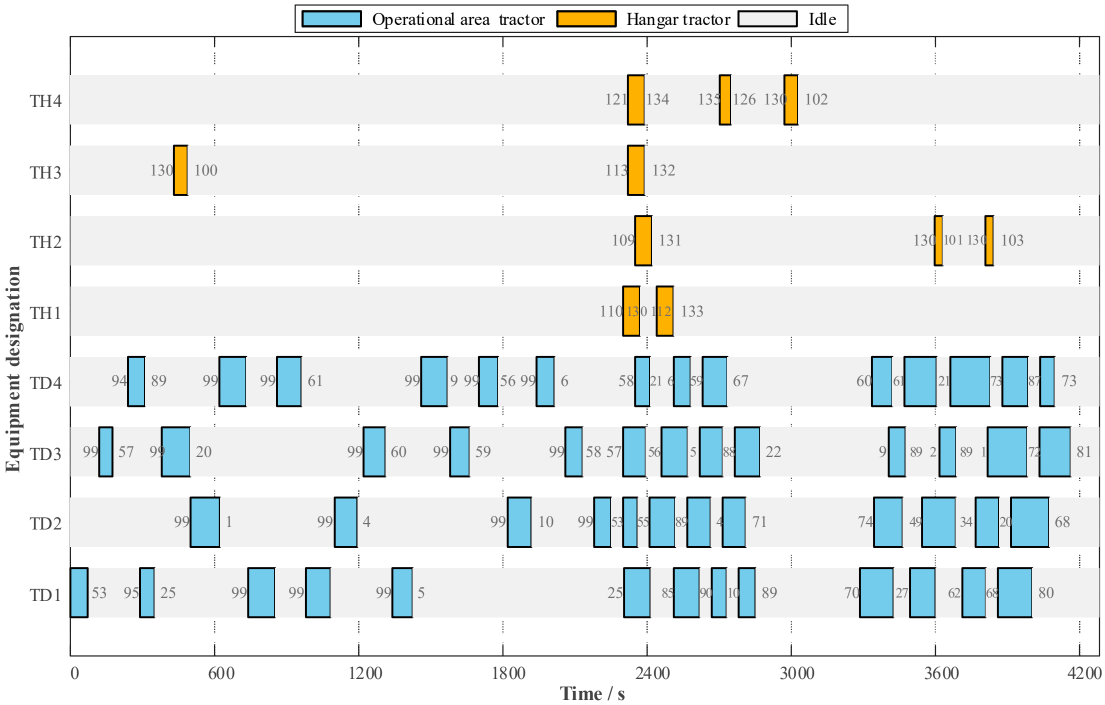


Figure 13. Gantt chart of tractor scheduling.

In summary, the scheduling scheme generated by the EDGC-ALNS algorithm exhibits tight connections among all phases, strict adherence to formation priorities, and reasonable allocation of tractor and elevator resources, validating the effectiveness and practicality of the model and algorithm in complex-scale scenarios.

8. Discussion

8.1. Main Results and Mechanism Analysis

This paper addresses the collaborative scheduling problem of staging and transfer operations for mixed UAV fleets preparing for re-launch at offshore wind power substations. An optimization model is constructed with objectives of minimizing total transfer time, configuration changes, and workload variance, and an EDGC-ALNS algorithm is designed. Simulation results show that the proposed algorithm achieves the optimal mean objective values and the lowest standard deviation across all six test cases.

The performance improvement of EDGC-ALNS mainly arises from three aspects. First, the event-driven configurable greedy initial solution generation strategy maintains the configuration–space joint state and generates a high-quality feasible solution at the initial scheduling stage, providing a favorable starting point for subsequent search. Second, the domain-knowledge-driven destruction and repair operators can make targeted adjustments to key conflicts such as parking spot competition, imbalanced tractor loads, and configuration switching, making the neighborhood search more problem-specific. Third, the adaptive weight mechanism and the simulated annealing acceptance criterion balance global exploration and local exploitation, helping the algorithm escape from local optima. The combined effect of these mechanisms enables the algorithm to maintain strong search capability under the deeply coupled configuration, space, and temporal constraints.

8.2. Comparison with Related Research

Compared with most existing studies that focus on single-phase or loosely coupled scheduling problems, the main difference with this paper lies in treating recovery, support, and pre-departure preparation as a continuously coupled process and in regarding configuration transition as an actively adjustable scheduling variable. Existing studies mostly adopt fixed-configuration assumptions or treat configuration merely as a static condition before operations, making it difficult to reflect the direct impact of configuration changes on parking spot compatibility, path passability, and temporal arrangement. The model in this paper integrates group priority, aircraft type differentiation, and configuration transition into a unified constraint system, thereby supplementing the deficiency of existing research in group–space coupling constraints.

In terms of the solution algorithm, this paper proposes the EDGC-ALNS algorithm. First, an event-driven configurable greedy initial solution generation strategy is designed to generate feasible initial solutions through an event-driven mechanism and configuration–space joint state maintenance. Second, domain-knowledge-driven destruction and repair operators are designed, and an adaptive weight mechanism together with a simulated annealing acceptance criterion is incorporated to effectively balance global exploration and local exploitation.

Simulation experiments validate the effectiveness of the model and algorithm from multiple dimensions. In terms of solution quality, EDGC-ALNS achieves the optimal mean objective values in all six test cases, with improvements of 7.0–7.9% over the conventional ALNS and 14.4–22.1% over the genetic algorithm, while also exhibiting the lowest standard deviation, demonstrating superior solution quality and stability compared with the comparison algorithms. In terms of computational efficiency, the algorithm achieves an average runtime of 10.1 s at the 20-UAV scale, approximately 20.7% shorter than that of the conventional ALNS. Although slightly higher than that of the genetic algorithm, it remains within an acceptable range, and the growth trend is generally consistent with the theoretical quadratic complexity. Statistical significance tests indicate that the differences between EDGC-ALNS and the two comparison algorithms are statistically significant, confirming that the performance improvement is not attributable to random fluctuations. The analysis of the scheduling scheme using Case 3 as an example demonstrates that the algorithm strictly adheres to group priorities, achieves reasonable allocation of tractor and elevator resources, and maintains tight connections among all phases, validating the effectiveness and practicality of the model and algorithm in complex-scale scenarios.

The proposed method can provide decision support for UAV logistics distribution scheduling at offshore wind power substations. It helps shorten pre-departure preparation time, reduce mechanism wear caused by frequent configuration switching, and balance the workload among transfer crews.

8.3. Influence of Environmental Disturbances on Deck Transfer Scheduling

This paper's scheduling model assumes that the deck layout and static obstacles remain unchanged during the scheduling period, and that the transfer times in the path library are deterministic constants, without explicitly considering environmental disturbances such as waves, wind, and currents. To evaluate the engineering rationality of this assumption, this section conducts a qualitative analysis from three aspects—influencing factors, influence magnitude, and engineering tractability—by drawing on studies of the structural dynamic response of offshore booster stations and motion compensation for offshore platforms.

In terms of influencing factors, the environmental loads on an offshore booster station mainly include wind loads, wave loads, and current loads. Reference [32] investigated the dynamic response of a certain 220 kV offshore booster station under the combined action of wind, waves, and currents based on a finite element model. The results showed that the natural frequencies of all modes of the offshore booster station lie outside the frequency range affected by wave resonance, so the structure will not resonate with waves; the dynamic response of the structure increases with the wave return period or typhoon wind level, among which wind load is the dominant load, while the responses induced by wave and current loads are relatively small. Reference [32] further pointed out that when only wave and current loads act, the stress and displacement responses of the lower steel piles are the largest, whereas the responses of the upper structure are very small. This is because wave and current loads act on the lower steel piles at the water surface and are difficult to transmit to the upper structure. For the deck transfer scheduling studied in this paper, this conclusion implies that the disturbances directly affecting the deck operation environment mainly originate from wind loads, while wave and current loads have already attenuated significantly by the time they are transmitted to the deck through the lower structure.

In terms of influence magnitude, the numerical results of Reference [32] showed that under the action of a Category 16 typhoon alone, the maximum resultant displacement of the upper structure of the offshore booster station is 11.24 cm; under the combined action of wind, waves, and currents, when the loads act perpendicularly to the structural axis plane, the structure produces the maximum stress and displacement responses, with a maximum resultant displacement of 23.06 cm. The wingspan of typical cargo unmanned aerial vehicles (UAVs) exceeds 15 m. According to the safety margin setting in Section 5.1, the transfer safety margin is sufficient to ensure the feasibility and safety of the path. References [33,34], through hydrodynamic analyses of floating wind turbine platforms, further found that the low-frequency motion period of semi-submersible platforms is typically much longer than the dominant wave period, and that platform motion is dominated by quasi-static offset, which remains within a controllable range under survival sea conditions. Moreover, four-column platforms exhibit superior dynamic response characteristics in surge, sway, pitch, and yaw directions, with more stable mooring tension variations. The above studies indicate that even on floating platforms, the spatial offset caused by platform motion mainly manifests as low-frequency, small-amplitude quasi-static displacement, rather than violent motion that alters the deck layout or path topology.

In terms of engineering tractability, Reference [35] addressed the trajectory tracking problem of an offshore platform manipulator under wave disturbances by constructing an AR model-based wave time series prediction and feedforward compensation control framework, and pointed out that there is a time delay in the transmission of wave disturbances from the platform to the manipulator, and that short-term prediction can control the prediction error rate within 0.7%. This conclusion has reference significance for deck transfer scheduling: environmental disturbances manifest at the scheduling level as predictable and compensable periodic disturbances. The time scale of the transfer tasks in this paper is on the order of hundreds to thousands of seconds, whereas the wave period is typically 10–15 s. Environmental disturbances manifest as high-frequency, small-amplitude disturbances at the scheduling scale, and a single period is far shorter than the duration of one transfer task, so they will not alter the sequence between tasks or the resource allocation pattern.

Based on the above analysis, the effects of platform motion and airflow disturbances on deck transfer scheduling are bounded and local. The spatial offset caused by platform motion is on the order of centimeters to decimeters, which can be absorbed by the safety margin preset in the path library. The time scale of environmental disturbances is far smaller than that of transfer tasks, so they manifest as high-frequency disturbances that can be smoothed at the scheduling decision granularity. The deterministic scheduling model adopted in this paper is therefore equivalent to conservative planning under the upper bound of disturbances.

8.4. Applicability of the Model Under Dynamic Disturbances and Rescheduling Potential

This paper focuses on deterministic scheduling under predefined assumptions. However, in practical operations, unexpected situations such as variations in transfer times, equipment failures, and changes in the priorities of sortie tasks may occur. This section provides a brief analysis of these situations and the corresponding countermeasures.

8.4.1. Applicability Analysis of Three Types of Dynamic Disturbances

The model proposed in this paper exhibits different degrees of applicability to the three types of dynamic disturbances.

(1) Variations in transfer times. As discussed in Section 8.3, minor fluctuations in transfer times (e.g., those caused by platform motion and environmental disturbances) manifest as high-frequency, small-amplitude disturbances at the scheduling scale. They can be absorbed by the safety margin preset in the path library and will not alter the sequence of tasks or the resource allocation pattern. Therefore, the current model possesses intrinsic robustness against such disturbances. However, if transfer times undergo substantial variations (e.g., a temporary path blockage leading to a significantly increased detour time), the temporal relationships after the variation point may become invalid, and subsequent transfer tasks need to be regenerated from the time of variation.

(2) Equipment failures. The current model assumes that tow tractors and elevators remain continuously available throughout the entire scheduling horizon, without considering equipment failures. Once a piece of equipment fails during task execution, its assigned but unfinished procedures cannot continue, and subsequent procedures dependent on that equipment are also affected. In this case, from the time of failure, the unfinished procedures undertaken by the affected equipment need to be released and reassigned to other available equipment.

(3) Changes in the priorities of sortie tasks. The current model treats group priorities as static input parameters, and all scheduling decisions are made based on fixed priorities. If priorities change temporarily, the original maintenance and staging sequence arranged according to the old priorities will no longer be reasonable. From the time of change, the scheduling sequence needs to be redetermined according to the new priorities.

In summary, the current model is applicable to minor variations in transfer times. For substantial variations in transfer times, equipment failures, and priority changes, subsequent transfer tasks need to be regenerated from the time of variation.

8.4.2. General Rescheduling Process for Unexpected Situations

The EDGC-ALNS algorithm adopted in this paper is well suited to the above rescheduling requirements in terms of its architecture. First, the event-driven mechanism processes three types of events in chronological order: transfer start, transfer end, and maintenance completion. Disturbances such as equipment failures and priority changes can be modeled as new event types inserted into the event queue, triggering local rescheduling without reconstructing the entire schedule. Second, the destruction–repair operators provide local adjustment means for rescheduling: the phase-related destruction operator can remove all subsequent procedures of the affected UAV as a whole, providing room for replanning the entire path from the breakpoint; the high-resource-occupancy destruction operator can reassign tasks from overloaded equipment, making it suitable for procedure transfer after equipment failures. Third, the adaptive weight mechanism dynamically adjusts operator weights according to the degree of objective improvement, automatically favoring search strategies suitable for the current disturbance type. Fourth, the simulated annealing acceptance criterion allows inferior solutions, helping the algorithm escape local optima after disturbances. Figure 14 presents the general rescheduling process based on EDGC-ALNS.

8.5. Impacts of Physical Deployment Delays and HIL Validation Framework

The EDGC-ALNS algorithm in this paper has been validated in the MATLAB numerical environment. To enhance the engineering application value of this research, this section further discusses the impacts of communication latency, sensor positioning errors, and real-time execution delays on the event-driven mechanism during physical deployment.

Regarding communication latency, the HIL experiments in Reference [36] demonstrated that the path planning algorithm can still be effectively executed under near-real communication conditions. Reference [37] theoretically studied the stability of event-triggered control systems under communication latency and proved that the event-driven mechanism only triggers decision updates when events occur; millisecond-level communication latency will not cause event loss or reversal of causal order, but at most results in a slight shift in the event processing instant, while the logical relationships between events remain unchanged. The measured results in Reference [38] showed that the general latency is at the millisecond level, and wireless communication latency within 50 ms mainly affects control accuracy rather than task-level scheduling logic. Regarding sensor positioning errors and real-time execution delays, Reference [39] achieved centimeter-level positioning accuracy based on RTK-GNSS and IMU sensor fusion positioning through an extended Kalman filter; the 6-DOF position fluctuation range of the UAV is ±10~17.5 cm, and the detection distance accuracy 1 s before the event is approximately 32.7 cm. Overall, the centimeter-level positioning error and millisecond-level execution delay are both at a relatively small magnitude compared with the transfer safety margin and the time scale of transfer tasks in this paper. Therefore, they will not cause structural failure of the event-driven mechanism.

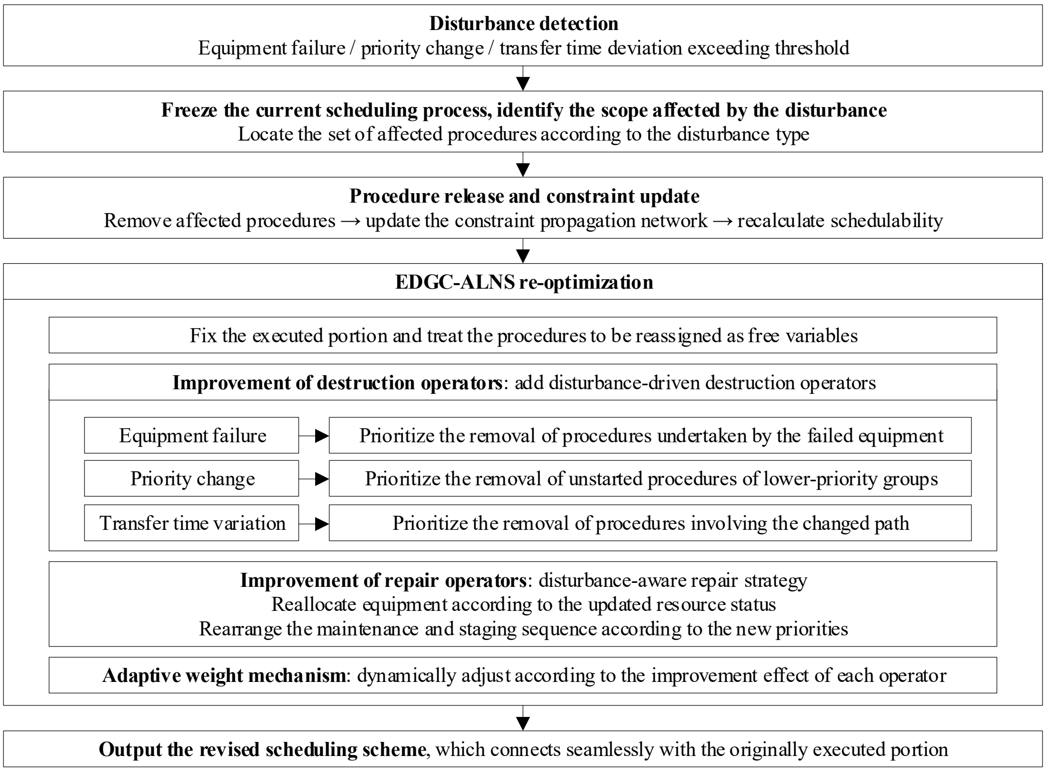


Figure 14. Rescheduling flowchart.

Based on the above analysis, this paper preliminarily proposes an HIL validation framework for the physical deployment of EDGC-ALNS, whose structure is shown in Figure 15. The algorithm side runs EDGC-ALNS and outputs scheduling instructions, which are transmitted through the ROS 2 or OPC UA protocol, with random delays injected into the channel. The instructions are received in a semi-physical or fully physical manner, and Gaussian white noise is injected into the sensor feedback channel to simulate positioning errors. The validation metrics include task completion time deviation, resource conflict count, and event processing timing deviation.

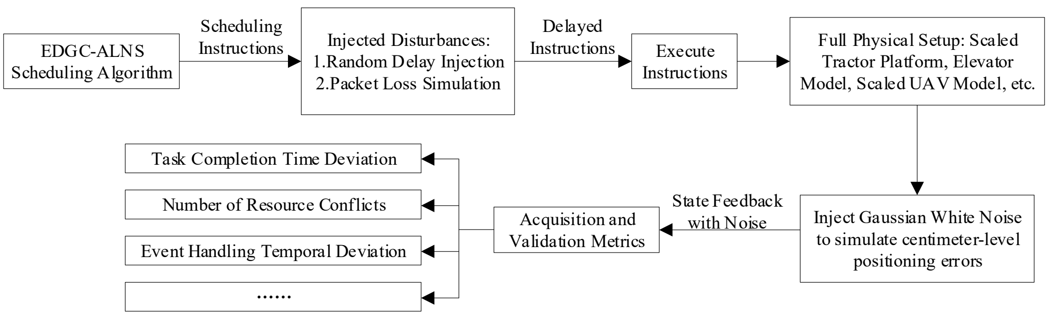


Figure 15. Flowchart of the HIL Verification Framework.

8.6. Limitations and Future Research Directions

This paper still has several limitations. The study focuses on the pre-departure preparation stage after a single recovery, and a complete rotation scheduling capability has not yet been formed. The problem has not been extended to multi-platform collaborative scheduling. The algorithm is validated only through numerical simulations, and hardware-in-the-loop or physical platform validation has not yet been conducted. Future research can address rescheduling under dynamic emergencies, cross-platform resource collaborative scheduling, and hardware-in-the-loop validation.

First, rescheduling under dynamic emergencies. In actual offshore operations, equipment failures, changes in task priorities, and unexpected interruptions may occur. Future research can investigate event-driven dynamic rescheduling models and online scheduling strategies to improve the robustness of scheduling schemes under disturbances. Specific directions include event-driven dynamic rescheduling models, online scheduling strategies considering changes in task priorities, and emergency resource allocation methods for equipment failures.

Second, cross-platform and cross-region resource collaborative scheduling. As the scale of unmanned aerial vehicle fleets and the scope of operations expand, resource scheduling within a single substation may become insufficient. Future research can investigate resource collaborative scheduling among multiple platforms and regions, including cross-platform sharing of tractors and elevators, collaborative launch of mixed unmanned aerial vehicle formations, and joint scheduling of multiple bases.

Third, hardware-in-the-loop and physical platform validation. The current algorithm is validated only in a numerical simulation environment. Future research can construct a hardware-in-the-loop simulation testbed, introduce factors such as sensor positioning errors, actuator response dynamics, communication delays, and real-time interruptions, and verify the feasibility of the algorithm under physical deployment conditions. Specific directions include a hardware-in-the-loop validation framework for scheduling algorithms, analysis of the impact of communication delays and sensor uncertainties, and optimization of event-driven mechanisms under real-time execution delays.

9. Conclusions

This paper addresses the collaborative scheduling problem of staging and transfer operations for mixed unmanned aerial vehicle fleets preparing for re-launch at offshore wind power substations. An optimization model is constructed that comprehensively considers group priority, aircraft type differentiation, and configuration transition factors, and an EDGC-ALNS algorithm is proposed to solve the problem. Simulation experiments show that the proposed model and algorithm outperform the conventional ALNS and the genetic algorithm in terms of solution quality, stability, and computational efficiency, and can provide effective decision support for unmanned aerial vehicle logistics distribution scheduling at offshore wind power substations. Future research can further consider rescheduling under dynamic emergencies, cross-platform resource collaborative scheduling, and physical platform validation.

Author Contributions: Conceptualization, W.H. and X.S.; methodology, X.L. and X.S.; software, X.L. and B.W.; validation, F.G. and C.L.; formal analysis, X.L. and W.H.; investigation, X.L.; resources, W.H. and F.G.; data curation, X.L.; writing—original draft preparation, X.L.; writing—review and editing, C.L. and W.H.; visualization, X.L. and B.W.; supervision, W.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Young Scientists Fund of the National Natural Science Foundation of China, grant number 62403486.

Data Availability Statement: Data are available upon reasonable request from the corresponding author.

Acknowledgments: The authors confirm that Generative AI (GenAI) with Deepseek R1 All was used solely for language translation assistance in the preparation of this manuscript. Specifically, GenAI was used to assist in translating the Chinese manuscript into English. It was not used in the study design, literature review, data analysis, algorithm development, numerical simulation, interpretation of results, or the generation of substantive research content. All research content, including the problem formulation, mathematical model, algorithm design, experimental results, and

conclusions, was independently completed by the authors. All GenAI-assisted translations were carefully reviewed, revised, and verified by the authors, who take full responsibility for the accuracy and integrity of the final manuscript.

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

References

1. Irawan, C.A.; Salhi, S.; Buticchi, G.; Chan, H.K. A Tandem Mother-Daughter-Vessel Routing Problem for Maintenance Activities in Offshore Wind Farms. *Transp. Res. Part E Logist. Transp. Rev.* **2026**, *211*, 104891. https://doi.org/10.1016/j.tre.2026.104891.
2. Schopferer, S.; Schitz, P.; Spiller, M.; Donkels, A.; Nagarajan, P.; Krause, F.; Schirmer, S.; Torens, C.; Dauer, J.C.; Cain, S.; et al. Offshore Wind Farm Delivery with Autonomous Drones: A Holistic View of System Architecture and Onboard Capabilities. *Drones* **2025**, *9*, 295. https://doi.org/10.3390/drones9040295.
3. Xiong, J.-J.; Zheng, E.-H. Position and Attitude Tracking Control for a Quadrotor UAV. *ISA Trans.* **2014**, *53*, 725–731. https://doi.org/10.1016/j.isatra.2014.01.004.
4. Chagas, G.O.; Coelho, L.C.; Laganà, D.; Beraldi, P. A Dynamic Drone Routing Problem with Uncertain Demand and Energy Consumption. *Transp. Res. Part B Methodol.* **2025**, *202*, 103335. https://doi.org/10.1016/j.trb.2025.103335.
5. Mannan, A.; Kahtan, H.; Mustafa, M.B.; Ahmad, R.; Abdulhak, M.; Atiquzzaman, M. Mobility Challenges and Issues in Unmanned Aerial Vehicle (UAV) Path Planning: A Systematic Review. *Comput. Netw.* **2025**, *273*, 111766. https://doi.org/10.1016/j.comnet.2025.111766.
6. Han, W.; Li, C.; Su, X.; Zhang, Y.; Guo, F.; Yu, T.; Li, X. Real-Time Decision Support for Bolter Recovery Safety: Long Short-Term Memory Network-Driven Aircraft Sequencing. *Def. Technol.* **2026**, *56*, 184–205. https://doi.org/10.1016/j.dt.2025.09.038.
7. Yu, L.; Zhu, C.; Shi, J.; Zhang, W. An Extended Flexible Job Shop Scheduling Model for Flight Deck Scheduling with Priority, Parallel Operations, and Sequence Flexibility. *Sci. Program.* **2017**, *2017*, 2463252. https://doi.org/10.1155/2017/2463252.
8. Peng, J.; Gao, W.; Li, Y.; He, S.; Wang, K.; Xu, M. A Heuristic Solution for Multi-Mode Resource-Constrained Project Scheduling Problem with Transfer Times: The Multi-Project Case. *Expert Syst. Appl.* **2025**, *290*, 128122. https://doi.org/10.1016/j.eswa.2025.128122.
9. Su, X.; Cui, R.; Li, C.; Han, W.; Liu, J. A Heuristic Solution Framework for the Resource-Constrained Multi-Aircraft Scheduling Problem with Transfer of Resources and Aircraft. *Expert Syst. Appl.* **2023**, *228*, 120430. https://doi.org/10.1016/j.eswa.2023.120430.
10. Liu, Y.; Han, W.; Su, X.; Cui, R. Optimization of Fixed Aviation Support Resource Station Configuration for Aircraft Carrier Based on Aircraft Dispatch Mission Scheduling. *Chin. J. Aeronaut.* **2023**, *36*, 127–138. https://doi.org/10.1016/j.cja.2022.06.023.
11. Chen, X.; Chen, Q.;. Luo, Y.; Wang, J.; Xu, M. Dynamic parallel scheduling of heterogeneous carrier-based aircraft deck support operations. *Acta Aeronaut. Astronaut. Sin.* **2025**, *46*, 244–263. https://doi.org/10.7527/S1000-6893.2024.31329.
12. Guo, F.; Han, W.; Liu, Y.; Liu, J.; Su, X.; Cheng, L. Scheduling for maintenance and service support of carrier-based aircraft based on variable operation process. *Acta Aeronaut. ET Astronaut. Sin.* **2025**, *46*, 531195. https://doi.org/10.7527/S1000-6893.2024.31195.
13. Jia, S.; Qi, N.; Li, Z.; Zhou, G.; Liu, Y. Mission Planning for UAV Swarm Air-Launched by Single Carrier Aircraft under Multiple Constraints. *Aerosp. Sci. Technol.* **2026**, *168*, 111375. https://doi.org/10.1016/j.ast.2025.111375.
14. Deng, Z.; Liu, X.; Dou, Y.; Su, X.; Li, H.; Wang, L.; Wang, X. Autonomous Sortie Scheduling for Carrier Aircraft Fleet under Towing Mode. *Def. Technol.* **2025**, *43*, 1–12. https://doi.org/10.1016/j.dt.2024.07.011.
15. Yoon, H.-C.; Oh, S.-H.; Woo, J.H.; Chung, J.-H.; Lee, H.; Jung, S.-A. Analysis of Sortie Generation on an Aircraft Carrier with Discrete Event Simulation. *Int. J. Nav. Archit. Ocean Eng.* **2025**, *17*, 100697. https://doi.org/10.1016/j.ijnaoe.2025.100697.
16. Yin, J.; Sun, B.; Fan, Y.; Shen, L.; Shi, Z. A Two-Tier Genetic Algorithm for Real-Time Virtual–Physical Fusion in Unmanned Carrier Aircraft Scheduling. *J. Mar. Sci. Eng.* **2025**, *13*, 856. https://doi.org/10.3390/jmse13050856.
17. Shi, H.; Wang, N.; Liu, Q. A Small-Sample Graph Neural Network Approach for Predicting Sortie Mission Reliability of Ship-borne Vehicle Layouts. *J. Mar. Sci. Eng.* **2026**, *14*, 599. https://doi.org/10.3390/jmse14070599.
18. Chen, C.; Demir, E.; Huang, Y. An Adaptive Large Neighborhood Search Heuristic for the Vehicle Routing Problem with Time Windows and Delivery Robots. *Eur. J. Oper. Res.* **2021**, *294*, 1164–1180. https://doi.org/10.1016/j.ejor.2021.02.027.
19. Gao, X.; Liu, D.; Shan, C. ALNS-TS based fast optimization algorithm for large-scale maintenance task scheduling. *CIESC J.* **2023**, *74*, 4645–4655. https://doi.org/10.11949/0438-1157.20230883.
20. Koyuncuoğlu, M.U.; Demir, L. Buffer Capacity Allocation in Unreliable Production Lines: An Adaptive Large Neighborhood Search Approach. *Eng. Sci. Technol. Int. J.* **2021**, *24*, 299–309. https://doi.org/10.1016/j.jestch.2020.08.012.

21. Lei, Q.; Gao, Y.; Zhou, Y.; Wu, Z. Multi-delivery option path planning based on improved ALNS algorithm. *Syst. Eng. Electron.* **2025**, *47*, 173–181.
22. Mao, C.; Hu, G.; Gu, Y.; Wu, G. Low Earth Orbit Early Warning Satellite Scheduling Algorithm based on Improved Adaptive Large Neighborhood Search. *Control Theory Appl.* **2026**, 1–15.
23. Li, Y.; Luo, X.; Yin, X.; Chen, Y. Optimization of Truck-Drone Collaborative Delivery Route Considering Dynamic Synchronization. *J. South China Univ. Technol. (Nat. Sci. Ed.)* **2026**, *54*, 1–11. https://doi.org/10.12141/j.issn.1000-565X.250275.
24. Westheider, J.; Rückin, J.; Popović, M. Multi-UAV Adaptive Path Planning Using Deep Reinforcement Learning. In Proceedings of the 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Detroit, MI, USA, 1–5 October 2023; pp. 649–656. https://doi.org/10.48550/arXiv.2303.01150.
25. Chen, K.; Liu, X.; Chun, J.; Cao, S.; Chen, Y. Model and algorithm for scheduling imaging satellite constellations based on multi-type tasks. *Control Decis.* **2025**, *40*, 1913–1921. https://doi.org/10.13195/j.kzyjc.2024.0897.
26. Hendel, G. Adaptive Large Neighborhood Search for Mixed Integer Programming. *Math. Program. Comput.* **2022**, *14*, 185–221. https://doi.org/10.1007/s12532-021-00209-7.
27. Keskin, M.; Çatay, B.; Laporte, G. A Simulation-Based Heuristic for the Electric Vehicle Routing Problem with Time Windows and Stochastic Waiting Times at Recharging Stations. *Comput. Oper. Res.* **2021**, *125*, 105060. https://doi.org/10.1016/j.cor.2020.105060.
28. Li, K.; Li, X.; Wu, Y.; Deng, Z.; Wang, Y.; Meng, Y.; Li, B.; Su, X.; Wang, L.; Wang, X. Autonomous Dispatch Trajectory Planning of Carrier-Based Vehicles: An Iterative Safe Dispatch Corridor Framework. *Def. Technol.* **2026**, *56*, 83–95. https://doi.org/10.1016/j.dt.2025.09.006.
29. Gugan, G.; Haque, A. Path Planning for Autonomous Drones: Challenges and Future Directions. *Drones* **2023**, *7*, 169. https://doi.org/10.3390/drones7030169.
30. Baharuddin, A.; Basri, M.A.M. Self-Tuning PID Controller for Quadcopter Using Fuzzy Logic. *Int. J. Robot. Control Syst.* **2023**, *3*, 728–748. https://doi.org/10.31763/ijrcs.v3i4.1127.
31. Wang, Z.; Yuan, Q.; Zhu, Y.; Hu, Y.; Chen, H.; Xie, X.; Gu, W. Fast and Intelligent Proportional–Integral–Derivative (PID) Attitude Control of Quadrotor and Dual-Rotor Coaxial Unmanned Aerial Vehicles (UAVs) Based on All-True Composite Motion. *Drones* **2024**, *8*, 747. https://doi.org/10.3390/drones8120747.
32. Xu, D.; Zhang, H.; Tao, A.; Wang, K.; Su, K. Study on Structural Dynamic Response of Offshore Booster Station under Combined Wind, Wave and Current Loads. *Water Power* **2022**, *48*, 72–79. https://doi.org/10.3969/j.issn.0559-9342.2022.08.014.
33. Yang, S.; Li, Y.; Wang, Z.; Zhao, Z.; Wang, J.; Zhou, L. An Optimized Four-Float Semi-Submersible Offshore Wind Turbine Platform: Hydrodynamic and Motion Response Evaluation. *J. Mar. Sci. Eng.* **2026**, *14*, 807. https://doi.org/10.3390/jmse14090807.
34. Zhang, J.; Yue, F.; Sun, K.; Kang, W.-H.; Shang, S.; Chao, Z.; Wei, M.; Liu, Z. Dynamic Response Analysis of Innovative Concrete Offshore Floating Wind Turbine Platforms. *Results Eng.* **2025**, *28*, 108264. https://doi.org/10.1016/j.rineng.2025.108264.
35. Liu, J. *Sliding Mode Control Strategy for Offshore Platform Robotic Arms with Disturbance Estimation*; Dalian University of Technology: Dalian, China, 2026. https://doi.org/10.26991/d.cnki.gdllu.2025.003014.
36. Yue, W.; Li, B.; Zhang, X.; Yang, Q. HDFOS: Path Planning for Heterogeneous Truck-Drone Cooperative Delivery and Pickup Services. *Intell. Serv. Robot.* **2025**, *18*, 775–798. https://doi.org/10.1007/s11370-025-00615-z.
37. Zhang, K.; Gharesifard, B.; Braverman, E. A Note on Stability of Event-Triggered Control Systems With Time Delays. *IEEE Trans. Autom. Control* **2024**, *69*, 2031–2035. https://doi.org/10.1109/TAC.2023.3321698.
38. Lyu, H.; Pang, Z.; Bhimavarapu, K.; Yang, G. Impacts of Wireless on Robot Control: The Network Hardware-in-the-Loop Simulation Framework and Real-Life Comparisons. *IEEE Trans. Ind. Inform.* **2023**, *19*, 9255–9265. https://doi.org/10.1109/TII.2022.3227639.
39. Nguyen, P.D.; Isogai, R.; Kondou, K.; Shoji, Y. Prediction of Ultra-High-Speed Spots Using RTK-GNSS Sensor Fusion for UAV-to-UAV mmWave/THz Communications. *IEEE Access* **2025**, *13*, 106942–106957. https://doi.org/10.1109/ACCESS.2025.3580781.