drones

Article

Automated Sortie Scheduling Optimization for Fixed-Wing Unmanned Carrier Aircraft and Unmanned Carrier Helicopter Mixed Fleet Based on Offshore Platform

Zixuan Liu [1], Wei Han [1], Yu Wu [2,*], Xichao Su [3,*] and Fang Guo [1]

[1] Aviation Foundation College, Naval Aviation University, Yantai 264001, China
[2] College of Aerospace Engineering, Chongqing University, Chongqing 400044, China
[3] School of Aviation Operation and Support, Naval Aviation University, Yantai 264001, China
* Correspondence: cquwuyu@cqu.edu.cn (Y.W.); suxich@126.com (X.S.)

Abstract: Optimizing naval aircraft deck sortie scheduling is important for improving the sortie efficiency of naval aircraft groups. It is also an important link in realizing the automated scheduling of aircraft carriers. Firstly, a single-aircraft surface transit path library was constructed. This paper used the improved A* algorithm and the optimal control deck path planning algorithm to construct the transit path library between the deck parking position and take-off position, or between warmable and non-warmable parking positions. Secondly, a mathematical optimization model was constructed for the sortie scheduling of a mixed fleet of fixed-wing carrier aircraft and carrier helicopters. The model took the maximum sortie time of the fleet and taxiing time of the fleet of fixed-wing aircraft as the optimization objectives, and considered the process flow, spatial, and resource constraints. Thirdly, a discretized improved whale optimization algorithm (IWOA) was designed to solve the problem. To improve the optimization-seeking capability of the algorithm, we discretized the algorithm, optimized its coding method, and introduced pre-constraints, learning factors, parameter improvements, and population restart. Finally, we set up various cases and proposed a new strategy for simulation experiments: these simulations verified the validity of the model, the excellent optimization performance of the IWOA, and the superiority of the sortie strategy. The research in this study will help to implement automated aircraft carrier scheduling.

Keywords: unmanned carrier aircraft; mixed fleet; sortie scheduling; improved whale optimization algorithm

check for updates

Citation: Liu, Z.; Han, W.; Wu, Y.; Su, X.; Guo, F. Automated Sortie Scheduling Optimization for Fixed-Wing Unmanned Carrier Aircraft and Unmanned Carrier Helicopter Mixed Fleet Based on Offshore Platform. *Drones* **2022**, *6*, 375. https://doi.org/10.3390/drones6120375

Academic Editor: Mostafa Hassanalian

Received: 27 October 2022
Accepted: 22 November 2022
Published: 23 November 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

The aircraft carrier fleet is a "great weapon" for maritime operations, as a compact maritime airfield containing many advanced technologies: its comprehensive effectiveness is closely related to the technical and tactical performance of its carrier aircraft, and to the efficiency of its deck operations, which are mainly responsible for planning and commanding the landings and sorties of aircraft, reflecting the "soft" strength of the carrier's integrated support. Compared to land-based airports, aircraft carrier decks have limited space, high safety risks for moving platform operations, a variety of tasks, complex processes and resource constraints, complex coupling of different operations, and demanding support timelines; therefore, researching carrier aircraft surface operations scheduling, so that the fleet can safely, collaboratively, and efficiently complete operational tasks such as take-off, landing, and surface support, is an important part of improving the effectiveness of carrier sorties, and a key technology for enhancing the comprehensive support and combat capability of aircraft carriers, and for automating their dispatch.

When a fleet of aircraft is deployed, the choice of support positions and take-off positions, the type of aircraft departing, and the sequence of deployments will greatly impact the sortie time; moreover, a large number of aircraft concentrated in a small space for mass

transfer is highly susceptible to collision, which can lead to significant conflict between safety and efficiency; furthermore, as unmanned technology continues to develop, unmanned aerial vehicles will become part of the Navy's carrier operating environment: this will add significant complexity to an already highly stressful and dangerous environment. The shift to a manned–unmanned shared environment means that aircraft, tow vehicles, and crew planning and scheduling must be more automated. Fully and rationally utilizing the deck space to ensure that the carrier aircraft can be deployed and depart quickly, safely, and in an orderly manner, and with a high degree of automation, is an important problem facing aircraft carriers today.

Current research on surface scheduling is focused on full-process scheduling and phased scheduling. In terms of model construction, for full process scheduling, Michini [1] and Ryan [2–4] conducted a series of studies based on methods such as integer programming, which paved the way for a holistic architecture for surface scheduling research. In phased scheduling, there are simulation models [5], queuing network models [6], intelligent body collaboration models [7], optimal control models [8], mixed-integer planning models [9], mission planning models [10], and mixed integer planning models [11]: they solve deck, munitions support, and sortie and recovery scheduling problems.

These reasonable models have achieved optimal or exact solutions, but they still lack adaptability to the problems investigated in this study: for example, the integer programming model is more suitable for the exact solutions to single-stage problems, and is extremely inefficient for the large-scale dispatching problem studied herein. Most other simulation and queueing models need human intervention, and cannot achieve full scheduling automation. Some of the models are too abstract, and not comprehensive enough to describe the scheduling problems in this study, or they are limited to simulating the sorties of fixed-wing aircraft, and lack careful consideration of mixed fleets of fixed-wing aircraft and carrier helicopters: for example, fixed-wing aircraft need to warm up their engines before sorties, but some deck spaces cannot be warmed up because of the structure of the aircraft carrier, so the fixed-wing aircraft need to be transferred first, while the carrier helicopters do not require this step; fixed-wing aircraft need to skid when taking off, which takes up a lot of deck space, while carrier helicopters only take up a small space at the take-off position; the fleet will interfere, owing to limitations in transfer equipment, space resources, etc. A new optimization model for the automated sortie scheduling problem for fixed-wing aircraft and helicopter mixed fleets is required: we refer to our model study as Automated sortie Scheduling Problem for Mixed Fleet (ASPMF).

The scheduling optimization algorithms are mainly exact solution methods, represented by the branch-and-bound methods: for example, the Gurobi commercial optimizer is based on the MIP model, but its application to complex constrained scheduling problem cases is less frequent. Although constrained planning methods use constraint propagation and neighborhood reduction, they possess the same computational complexity as the exact solution methods. Construction heuristics and meta-heuristic search algorithms based on heuristic rules (such as SA), greedy iterative algorithms based on neighborhood search, and evolutionary algorithms based on population intelligence search (such as GA, ABC, PSO, and quantum algorithms [12,13]) are computationally efficient, but do not easily obtain globally optimal solutions, being prone to premature maturity or restricted to local optima. Reinforcement-learning-based scheduling algorithms are another option, for which there is currently a lack of large amounts of data for training. In this study, an improved whale optimization algorithm (IWOA) with higher solution efficiency was designed, to solve the model optimally, and to realize the co-optimization of fleet deployment, sortie timing, and take-off position selection.

The fixed-wing carrier aircraft and carrier helicopters mentioned are both unmanned aircraft. The remainder of this section outlines the focus of work in this study.

Before conducting the scheduling optimization study, extensive preparatory work was done for this paper. We constructed an ASPMF path library. To perform single-carrier aircraft path planning, we used a carrier deck surface path planning algorithm that

combined the A* algorithm with an optimal control algorithm [8,14,15]: thus, we obtained the shortest path for the deck transfer of the carrier aircraft, which satisfied kinematic, terminal position constraints, and deck space resource constraints. An ASPMF path library of over 300 paths was constructed via route planning for any two aircraft positions.

In the model construction, we used the ASPMF path library to optimally model the ASPMF. In terms of the optimization objectives, we set up two optimization objectives, minimizing fleet departure time and carrier aircraft taxiing time, which were optimized hierarchically. In terms of constraints, we considered the process flow, space resource, and support resource constraints on the process of the carrier aircraft sortie. This model allowed us to solve multiple types of carrier aircraft sortie schedule problems that had never been studied.

In terms of the optimization algorithm, we proposed the improved whale optimization algorithm (IWOA) [16]. We discretely designed the algorithm using list coding, and improved it through a learning factor and pre-constraint, parameter dynamic improvement, and population restart.

Three sortie scenarios, for case simulations with 8, 12, and 16 aircraft, were constructed. Simulations were conducted to verify the validity of the model and the excellent optimization performance of the IWOA. A hybrid sortie strategy was proposed and compared with the existing sortie strategy, to demonstrate that it could effectively improve the efficiency of the sortie.

2. Related Works

2.1. Fleet Sortie Scheduling

The complete carrier aircraft take-off and landing operation flow is shown in Figure 1:

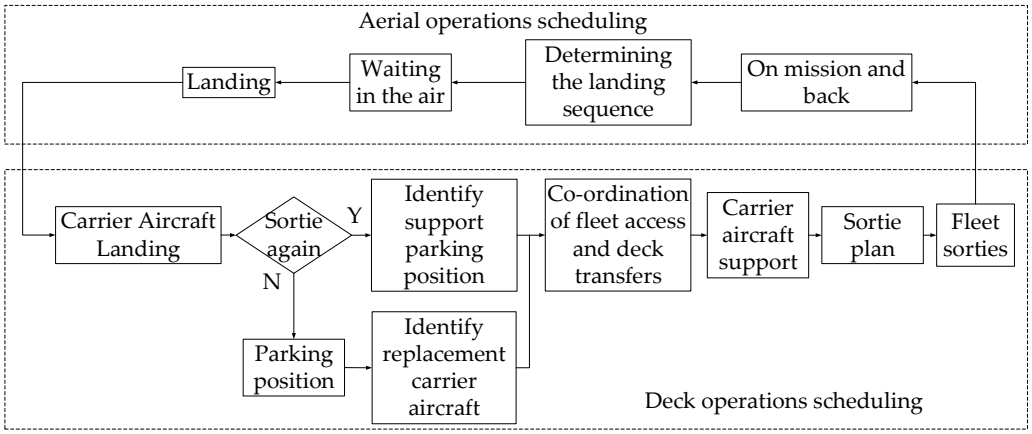


Figure 1. Carrier aircraft fleet take-off and landing operation flow.

The operation process includes air and surface operations, and its scheduling optimization is divided into surface and air operation scheduling. The entire operation process—except for air execution, air waiting, and landing—and the rest of the tasks in each phase, are the problems that need to be studied for the take-off and landing of carrier aircraft.

In recent years, several studies have been conducted on ship surface schedules. With regard to deck operations, in [17], the authors developed a robust scheduling model for flight deck operations, and designed a base hybrid teaching optimization algorithm that successfully improved the robustness of flight deck operations; in [5] the authors developed an integrated optimization model for job scheduling and resource allocation, and proposed a multi-objective super heuristic selection function-based model-solving method, to develop an effective job scheduling plan under different resource configurations. With regard to weapons transfer, in [11] the authors developed a mixed-integer programming model, and proposed a dual population immunity algorithm that could effectively cope with the problem of securing weapons for carrier aircraft under different sortie missions.

With regard to deck resource scheduling, in [18] the authors proposed an improved variable neighborhood search algorithm, with four original neighborhood structures for the joint optimization of flight deck resource station configurations and carrier pre-flight scheduling. With regard to fleet recovery, in [9,10] the authors formulated the sequencing problem of aircraft fleet landing, as an optimization problem with a cost function, and proposed a dynamic sequencing algorithm based on an ant colony algorithm, to obtain the optimal sequence of aircraft fleet landing. The authors then proposed a new distributed mission planning architecture that considered realistic constraints at all levels of the distributed structure (vortex effects, crowd effects, aircraft motion), and a distributed path planning algorithm based on an asynchronous planning strategy, to solve the carrier aircraft sortie mission planning problem. With regard to path planning, in [19] the authors developed a conceptual model of helicopter landing, containing two phases—approach and landing—and proposed a multiphase path planning algorithm with pigeon-inspired optimization (MPPIO), to solve the path planning problem of autonomous helicopters landing on aircraft carriers. In [20] the authors developed a path planning model that combined manual experience with an improved artificial swarm algorithm, to propose a method for planning the trajectory of a carrier aircraft on deck. In [21] the authors proposed a genetic algorithm and threat prediction-based probabilistic neural network (PNN) approach for real-time trajectory planning, which successfully implemented online flight trajectory planning. In [22] the authors designed a multihabitat parallel chaotic algorithm (KCMPSO) to solve the aircraft transfer path planning problem for aircraft carriers. As mentioned above, despite extensive research in the field of deck support, the study of fleet sortie scheduling has been mainly limited to path planning and fixed-wing aircraft. Path planning is extremely limited in its consideration of practical constraints, and we need a realistic, accurate, and efficient model and algorithm to solve the problem of scheduling the sorties of multiple types of unmanned carrier aircraft under complex constraints.

If we ignore the spatial constraints of the transfer paths and parking spaces, and consider the aircraft as workpieces—where warm-up, transfer, and take-off are operational processes in different phases, and transfer equipment in each phase as parallel machines—then we can abstract the ASPMF as a hybrid flow shop scheduling problem (HFSP), where the operational phases interfere with each other [23]; therefore, we implemented model improvements based on the hybrid flow shop model, to provide an in-depth study of DSPMF under complex constraints.

2.2. Hybrid Flow Shop Problem

Workshop scheduling problems include various classifications, such as single machine scheduling [24], parallel machine scheduling [25], flow shop scheduling [23], job shop scheduling [26], flexible shop scheduling [27], and open shop scheduling [28]. Depending on the type of parallel machine, HFSPs can be divided into three categories: identical parallel machine [29]; uniform parallel machine [30]; and uncorrelated parallel machine [31]. According to the abstraction method described in the previous section, the safeguard equipment involved in the process of carrier aircraft sorties can be abstracted as parallel machines, even if the individual tractors have the same function, because the waiting time to call the tractors is different. We can consider parallel machines to have the same function but different processing times; therefore, the problem we study can be abstracted as a uniform parallel machine problem.

Combined with the characteristics of our study, this involves several classifications of HFSPs. For example, only one carrier aircraft can take off from any take-off position, so the take-off operation of the carrier aircraft is a blocked HFSP [32]; there is a waiting position for the fixed-wing carrier aircraft after its take-off position, which is equivalent to a buffer zone, and this stage is a limited buffer HFSP [33]; compared to a fixed-wing aircraft, a naval helicopter lacks a warm-up process, so it is a skippable HFSP [34]; the warm-up process of a fixed-wing aircraft requires immediate taxiing when the warm-up is completed in order to reduce fuel consumption, and this phase is a zero-idle HFSP [35]. Therefore, although

the problem we studied was similar to the hybrid flow shop scheduling problem, it was more complex, and involved more constraints.

There are currently four main approaches to finding the optimal solution for HFSPs: integer programming, heuristic algorithms, meta-heuristic algorithms, and learning algorithms [36,37]. Traditional integer programming methods are only suitable for the exact solutions of small-scale problems derived from two-stage HFSPs to k-stage HFSPs, where complex constraints between stages and machines reduce the speed of integer programming solutions, making it difficult to find the optimal solution in an efficient time.

To solve two-stage HFSPs, researchers have used an exact algorithm to verify the derived lower bound for the problem. In [38], the authors formulated a mixed-integer programming model for an energy-efficient HFSP, and adopted an estimation of distribution algorithm to solve large scale problems; in [39], the authors formulated a new mixed-integer programming model for this multi-objective HFSP; to realize the green scheduling, an energy conservation/noise reduction strategy is embedded into this model.

In heuristic algorithms, scholars determine optimal solutions using assignment rules and local search mechanisms. In [40], the authors investigated a skippable HFSP with a minimum completion time as the optimization objective. The authors optimized the completion time by improving the LPT and SPT rules. In [41], the authors proposed heuristic algorithms along with priority rules based on traditional SPT and LPT rules, to solve a new variant of the HFSP.

In terms of meta-heuristic algorithms, in [42], the authors proposed a hybrid genetic algorithm-based ant colony optimization (GACO), to solve the minimum completion time problem for a re-entrant HFSP with time window constraints. In [43], the authors proposed an improved Particle Swarm Optimization (PSO) algorithm, to solve the HFSP for the same parallel machine, considering the influence of human factors. In [44], the authors constructed a hybrid biogeography-based optimization with a variable neighborhood search mechanism, to solve the no-wait HFSP. In [31], the authors proposed an improved genetic algorithm for solving the HFSP.

In terms of learning algorithms, in [45], the authors approximated the value function using a neural network to solve the flow shop scheduling problem; under the reinforcement learning framework, satisfactory results were obtained for all seven machine sets.

Except for a few individual problems, most workshop scheduling problems are NP-hard problems. Only intelligent scheduling methods are available for mid- and large-scale problems, to obtain a satisfactory solution that weighs computational timeliness and scheduling optimality.

3. Description of the Problem

The departure of fixed-wing naval aircraft consists of four main stages, which are outlined below.

1. Warm-up and self-test. According to the requirements of fixed-wing carrier engines, a warm-up and self-test are required before take-off, to confirm whether the performance indicators of the engine are normal. This stage generally requires 2–3 min. Because of the design characteristics of the carrier structure, some positions cannot operate aircraft warm-ups, as the fixed-wing aircraft tail flames would cause ablation of the carrier structure: therefore, some aircraft need to be transferred at this stage.
2. Waiting behind the deflector. When the aircraft completes the engine warm-up and self-test, it is transferred to the take-off position. If the target take-off position already exists, the aircraft will be transferred to the waiting position behind the target take-off position: as this stage involves deck transit, transit collision avoidance constraints must be considered, to prevent collisions due to interference in the path of the carrier aircraft.
3. Preparation at the take-off position. Once the aircraft has reached the target take-off position, it can be prepared for take-off.
4. Take-off. When the carrier aircraft has completed all preparations, it waits for the take-off order, and completes it.

For carrier helicopters, we divided the sortie process into two sub-stages, which are outlined below.

1. Transfer to take-off position. After the helicopter has completed its support, it can be transferred to the target take-off position: this stage requires consideration of the transit path blocking, aircraft position interference, and transit collision avoidance constraints. The road conditions determine the timing of the helicopter transfer, and the helicopter must wait in the original parking position when there is a helicopter in the target position.
2. Take-off preparation and take-off. Once the helicopter has reached the target take-off position, preparations are made for take-off, such as rotor deployment, navigation calibration, and control system self-testing. After completing this series of tasks, the helicopter can wait for the take-off command to execute the take-off, which takes approximately 5 min.

The complete take-off process is shown in Figure 2:

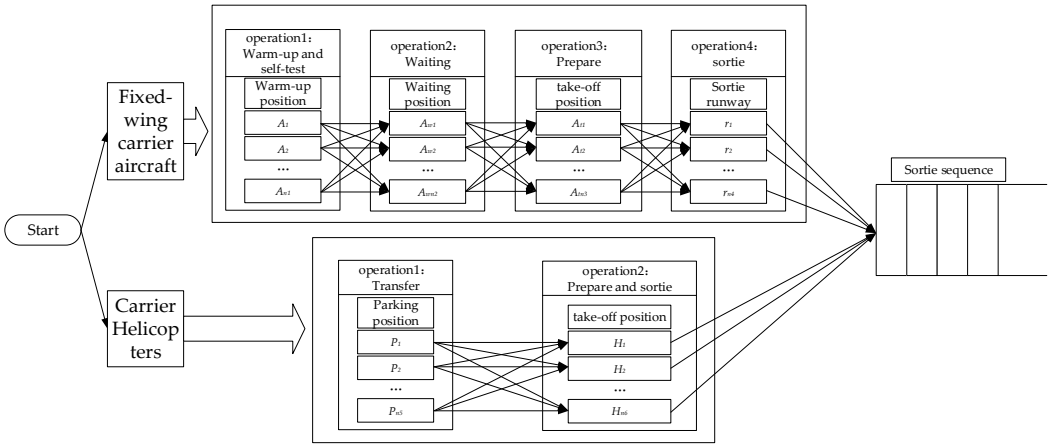


Figure 2. Process of aircraft sortie on deck.

A detailed analysis of the mixer group sortie process includes the constraints outlined below.

1. Warm-up position constraints. As mentioned above, a fixed-wing aircraft needs to be transferred first, for aircraft that are unable to complete engine warm-up in their initial position. The transfer must be operated when no warm-up position is available, after the other aircraft has left the parking position.
2. Transport collision avoidance restraints, to ensure that no collisions occur during the carrier aircraft transfer process.
3. Blockage of transit path by parking position. As shown in Figure 3, for helicopter take-off position C7, all helicopters selecting the C7 take-off position will interfere with the carrier aircraft A12, A13, and A14. Only after the A12, A13, and A14 carrier aircraft take off will the taxi to take-off position for C7 be clear.
4. Runway incursion constraint. As shown in Figure 3, if a carrier aircraft is parked at the C1 take-off position or the waiting position behind it, it will invade the take-off runway of the C3 take-off position, which will cause C3 to be unable to take off. Similarly, the carrier helicopters parked at the C4 take-off position occupy the runways corresponding to the C1, C2, and C3 take-off positions, and the carrier helicopters parked at the C6 take-off position occupy the runway corresponding to the C3 take-off position.
5. Take-off interval constraint. After a fixed-wing carrier aircraft takes off, its take-off wake disturbs the airflow field on the carrier surface, which leads to a higher risk for subsequent aircraft take-off, and the deflector plate behind the fixed-wing carrier aircraft take-off position also needs to be reset and cooled. For helicopters, it takes

time to reach a safe distance from the carrier after take-off; therefore, after a carrier aircraft has taken off, a period of time must be allowed for the next carrier aircraft to take off, to ensure safe sortie.

6. Operational flow constraint. Fixed-wing aircraft and carrier helicopters are required to complete their respective take-off procedures in a specific sequence before they can sortie.
7. Take-off weight constraint. When the take-off weight of the carrier aircraft exceeds a certain limit, the take-off must be performed at the C3 take-off position with a longer glide distance, to ensure that the carrier aircraft achieves the required speed when leaving the deck.
8. Take-off position selection constraint. As fixed-wing aircraft and helicopters have different types of take-off positions, they need to be constrained in the selection of take-off positions.

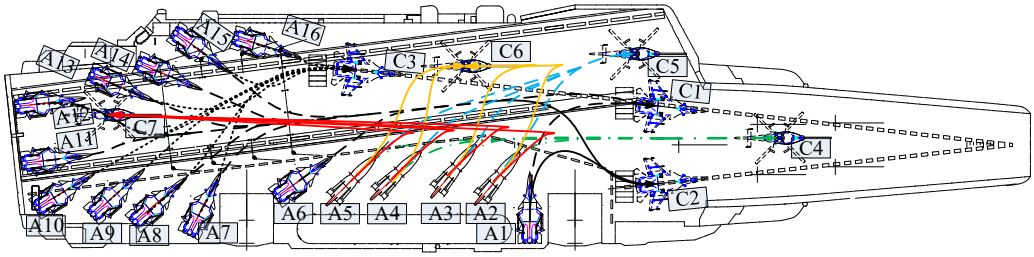


Figure 3. The sortie of a mixed fleet of fixed-wing carrier aircraft and carrier helicopters.

We consider that the wings of both the fixed-wing aircraft and the helicopters are folded during the transit stage, and are not deployed until they reach the take-off position.

In summary, ASPMF is a study to optimize the timing and positioning of a mixed fleet of aircraft at each stage of departure, under complex constraints such as warm-up, space resource, operational flow, and take-off position selection constraints, to minimize the completion time and deck taxiing time of each fleet of aircraft.

4. Path Library and Mathematical Formulation

4.1. Path Library Construction

To obtain the shortest path of ship transfer that satisfies the kinematic and start/end position attitude constraints, in order to improve the efficiency of ship transfer, we adopted an earlier reported path planning method [8,14,15], so as to carry out single aircraft deck path planning, and construct a ship deck transfer path library. The effect of single aircraft path planning is shown in Figure 4.

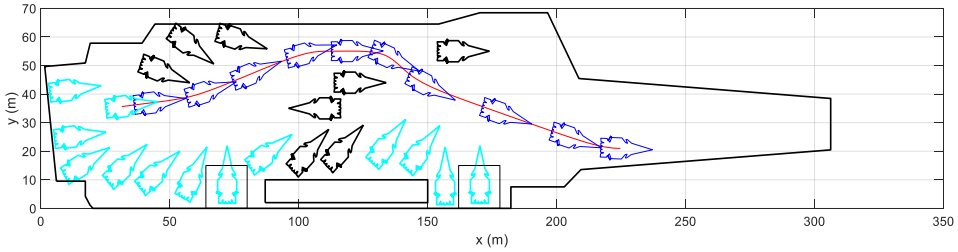


Figure 4. Single carrier aircraft path planning results.

The parameters in the optimal control algorithm were set as follows: obstacle shape parameter $p = 3$, aircraft front and rear wheelbase $L_1 = 5.88$ m, tractor front and rear wheelbase $L_2 = 2.4$ m, aircraft steering angle $\beta_{1\max} = \pi/4$, tractor speed $-1\ \mathrm{m/s} \leq v_2 \leq 1\ \mathrm{m/s}$, control quantity (tractor acceleration) $-1\ \mathrm{m/s^2} \leq u_2 \leq 1\ \mathrm{m/s^2}$, the angle between the axial direction of the carrier aircraft and the x-axis $-\pi \leq \theta_1 \leq \pi$, the angle between the axial direction of the tractor and the x-axis $-\pi \leq \theta_2 \leq \pi$, as the distance between the

tractor articulation point and the center of the rear wheels of the tractor was extremely small in the actual tractor $M_0 = 0$, and the control quantity $-1 \leq u_1 \leq 1$ (tractor steering angle $-\pi/4 \leq \beta_2 \leq \pi/4$). The transfer path bank between the deck and hangar stops is shown in Figure 5.

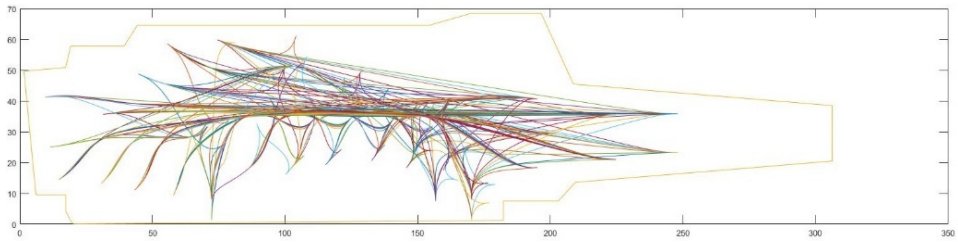


Figure 5. Path library.

4.2. Mathematical Formulation

4.2.1. Problem Assumptions

1. When an operation begins, it cannot be stopped;
2. Only one transfer equipment may be accepted for transfer at any transfer stage at any one time;
3. The carrier aircraft can perform a transfer operation with any transfer equipment corresponding to each stage;
4. The parameters of the path library are constant, and are unaffected by other transfer operations;
5. The interference of unexpected factors, such as malfunction, is not considered.

4.2.2. Notation

The relevant notations and decision variables for modeling the ASPMF are provided in Table 1.

Table 1. The relevant notations and decision variables for modeling the ASPMF.

Notation	Definition
I_1	The set of fixed-wing carrier aircraft $I_1 = \{1, 2, \cdots, Nf_1\}$.
I_2	The set of carrier helicopters $I_2 = \{1, 2, \cdots, Nf_2\}$.
I	The set of mixed fleet $I = I_1 \cup I_2 = \{1, 2, \cdots, Nf\}$, $Nf = Nf_1 + Nf_2$.
Ψ_j	The set of parking positions that can be used for the *j-th* stage of the start-up process.
P	The set of parking positions.
J	The set of sortie operations, $J = \{0, 1, 2, 3, 4\}$, 0 is the initial operation, the 1 operation of the helicopter is equivalent to the 2 operation of the fixed-wing carrier, and the 2 stage of the helicopter is equivalent to the 4 stage of the fixed-wing carrier.
R	The set of take-off positions $R = R_1 \cup R_2$,$R_1 = \{C1, C2, C3\}$, $R_2 = \{C4, C5, C6, C7\}$.
O_{ij}	The *j-th* operation of carrier aircraft *i-th*.
$TR(p, q, \boldsymbol{Ap}_t(p, q))$	The duration of the carrier aircraft transit from initial position p to target position q at time t.
tx_j	The duration of tethering and untethering for carrier aircraft.
dw_j	The duration of fixed-wing carrier aircraft *j-th* operation.
dh_j	The duration of carrier helicopter *j-th* operation.
γ_p	The take-off position corresponding to waiting position p,$p \in \Psi_2, \gamma_p \in \Psi_3$.
$\Phi_{\mathrm{p}}(p, q)$	The set of parking positions that interfere with the path from the initial position p to the target position q.

Table 1. *Cont.*

Notation	Definition
$\Phi_{\mathrm{r}}(p,q)$	The set of runways invaded by the path from the initial position p to the target position q.
δ_i	Wake interval of the *i-th* fixed-wing carrier aircraft or safe departure time of the *i-th* carrier helicopter.
Δtb	The duration of deflector cooling and resetting.
m_i	Take-off weight of the *i-th* fixed-wing carrier aircraft.
fm_p	Maximum take-off weight at take-off position p.
Sw_{ij}	The start time of the *j-th* support operation of the *i-th* carrier aircraft.
Ew_{ij}	The end time of the *j-th* support operation of the *i-th* carrier aircraft.
Sz_{ij}	The start time of the *j-th* transfer operation of the *i-th* carrier aircraft.
Ez_{ij}	The end time of the *j-th* transfer operation of the *i-th* carrier aircraft.
P_{ij}	The parking position occupied by the *j-th* operation of the *i-th* carrier aircraft.
Yp_{ijegp}	If the *j-th* operation of the *i-th* carrier aircraft occupies the same parking position as the *g-th* operation of the *e-th* carrier aircraft, and the *i-th* carrier aircraft has higher priority, then Yp_{ijegp} is 1; otherwise Yp_{ijegp} is 0.
Yr_{ijegr}	If the *j-th* operation of the *i-th* carrier aircraft invades the same take-off position as the *g-th* operation of the *e-th* carrier aircraft, and the *i-th* carrier aircraft has higher priority, then Yp_{ijegp} is 1; otherwise, it is 0.
x_{ie}	If the *i-th* carrier aircraft has a higher priority than the *e-th* carrier aircraft, then x_{ie} is 1; otherwise, it is 0.

4.2.3. Optimization Objectives

6. Minimize fleet sortie completion time

A carrier aircraft fleet generally operates in a cyclical cycle when conducting missions. Owing to the overly narrow deck of the carrier, it is necessary to complete the current wave of carrier aircraft sorties, to accelerate the wave cycle:

$$\min \mathrm{C}_{\max} = \max_{\forall i \in I} Ew_{i4} \tag{1}$$

7. Minimize fleet transit time

The shorter the transit time of the fleet, the safer the fleet transfer process:

$$\min TT = \sum_{i \in I} TR\left(P_{ij}, P_{i(j+1)}, \boldsymbol{Ap}_{Sz_{ij}}\left(P_{ij}, P_{i(j+1)}\right)\right),\ j = 2 \tag{2}$$

4.2.4. ASPMF Constraints

8. Timing constraint on workflow

Both fixed-wing aircraft and helicopters have multiple operations that are interlinked and have internal timing relationships. For transfer operations, the timing constraints are as follows:

$$Ez_{ij} = Sz_{ij} + \Delta Tz_{ij}, \forall i \in I, j \geq 1 \tag{3}$$

The transit time of the carrier aircraft ΔTz_{ij} is derived from the parking position of the two adjacent stages:

$$\Delta Tz_{ij} = \begin{cases} TR\left(P_{i(j-1)}, P_{ij}, \boldsymbol{Ap}_{Sz_{ij}}\left(P_{i(j-1)}, P_{ij}\right)\right) + tx_j, \text{ if } P_{i(j-1)} \neq P_{ij} \\ 0, \text{ otherwise} \end{cases} \tag{4}$$

support operational timing constraints as:

$$Ew_{ij} = Sw_{ij} + dw_j \text{ or } Ew_{ij} = Sw_{ij} + dh_j, \forall i \in I, j \geq 1 \tag{5}$$

constraints on transit operations and support operations as:

$$Sw_{ij} \geq Ez_{ij}, \forall i \in I, j \geq 1 \tag{6}$$

$$Sz_{ij} \geq Ew_{i(j-1)}, \forall i \in I, j \geq 2 \tag{7}$$

Equation (6) indicates that for both support and transfer operations of the same stage, the former must wait for the latter's completion before proceeding. Equation (7) indicates that transfer operations must not commence until the previous support stage has been completed.

In addition, a fixed-wing aircraft needs to wait for the deflectors to cool and reset before entering the take-off position:

$$Sz_{ej} \geq Yp_{ijejp}\left(Ew_{i(j+1)} + \Delta tb\right), \forall i, e \in I_1, j = 3, p \in P_j \tag{8}$$

9. Space resource constraint

This constraint consists mainly of a transit collision avoidance constraint, and occupancy of the same space priority constraint, for the former:

$$\begin{array}{l} \mathcal{A}_i\left(\boldsymbol{Road}_t\left(P_{i(j-1)}, P_{ij}\right)\right) \cap \mathcal{A}_e\left(\boldsymbol{Road}_t\left(P_{e(g-1)}, P_{eg}\right)\right) = \varnothing, \\ t \in \left[Sz_{ij}, Ez_{ij}\right] \cap \left[Sz_{eg}, Ez_{eg}\right], \forall i, e \in I, j, g \geq 1 \end{array} \tag{9}$$

The latter is mainly used to delay the transfer of the carrier aircraft when there is an interference between the transfer path and the take-off glide path of the fixed-wing carrier aircraft:

$$Sz_{ij} \leq Sz_{eg} + BM\left(1 - \max\left(Yp_{ijegp}, Yr_{ijegr}\right)\right), \forall i, e \in I, j, g \geq 1, \forall p \in P, \forall r \in R \tag{10}$$

10. Take-off position matching constraint

For fixed-wing aircraft, there are waiting positions behind each take-off position, and when a fixed-wing aircraft selects a waiting position, its take-off position is also determined:

$$P_{i(j+1)} = \gamma_{P_{ij}}, \forall i \in I, j = 2 \tag{11}$$

11. Take-off interval constraint

The wake interval should be met between successive sorties for fixed-wing carrier aircraft, and the safe departure interval should be met between successive sorties for carrier helicopters:

$$Sw_{ej} \geq x_{ie}\left(Ew_{ij} + \delta_i\right), \forall i, e \in I_1, j = 4 \tag{12}$$

12. Take-off weight constraint

Heavier fixed-wing aircraft can only take off from the C3 take-off position, so the choice of take-off position for fixed-wing aircraft needs to meet the weight requirements of the take-off position:

$$P_{ij} \in \left\{p \middle| m_i \leq fm_p\right\}, \forall i \in I_1, j = 3 \tag{13}$$

13. Decision variable constraint

Yp_{ijegp}, Yr_{ijegr}, and x_{ie} are Boolean 0–1 decision variables:

$$Yp_{ijegp},\ Yr_{ijegr},\ x_{ie} = \{0, 1\},\ \forall i,\ e \in I, j,\ g \geq 1 \tag{14}$$

14. Take-off position selection constraint

Because of the difference in take-off positions between fixed-wing aircraft and carrier helicopters, it is necessary to select the appropriate take-off positions according to the type

of carrier aircraft, which can be C1, C2, and C3 when the carrier aircraft is a fixed-wing aircraft, and C4, C5, C6, and C7 when the carrier aircraft is a carrier helicopter:

$$P_{ij} \in \begin{cases} R_1 \ i \in I_1, j = 3,4 \\ R_2 \ i \in I_2, j = 4 \end{cases} \tag{15}$$

5. IWOA

Based on the above modeling analysis, a solution to the ASPMF faced the following three difficulties compared to HFSPs: firstly, the start time of the transfer operation was not only constrained by the available transfer equipment, but also by the feasible path and multiple aircraft collision avoidance constraints; secondly, there was interference between the different operations of each aircraft, which was not a consistent flow direction of the HFSP; thirdly, the process flow of each type of aircraft was different.

Mimicking the feeding behavior of whales in nature, a novel population intelligence optimization algorithm called the WOA was proposed by Mirjalili in 2016 [16]. The WOA has a simple structure, requires few parameters to be adjusted, and is relatively easy to implement. The WOA has an information feedback mechanism and convergence factor that can be adjusted adaptively, to balance the local search with the global search. This section outlines how an IWOA, based on the standard WOA, was designed to solve the ASPMF.

5.1. The Principle of the WOA

The optimization steps of the algorithm were as follows: (1) initialize the parameters and individuals of the population; (2) calculate the fitness value of the individuals, and take the individual with the smallest fitness value as the optimum; (3) update the next generation population; (4) if the termination condition is reached, output the optimum individual; otherwise, return to step (3) to continue the optimization. There were three ways to update the population: encircling prey, the bubble-net attacking method, and searching for prey, which were the core of the algorithm.

5.2. Encoding and Decoding

In the coding stage, we assigned take-off positions C1, C2, and C3 or C4, C5, C6, and C7 according to the type of aircraft at a certain stop, and obtained $\overrightarrow{Y} = \left[y_1, y_2, \cdots, y_{Nf}\right]$, where y_j indicated the take-off position corresponding to the aircraft at parking position j; the take-off sequence of the aircraft was coded in a list, $\overrightarrow{X} = \left[x_1, x_2, \cdots, x_{Nf}\right]$, where x_j indicated the priority of the aircraft at stop j in the list, and the value range was $x_j \in \{1, 2, 3, \cdots, Nf\}$, the smaller the number, the higher the priority.

The final code was:

$$\overrightarrow{Z} = \overrightarrow{Y} + \frac{\overrightarrow{X}}{k} \quad k = 10, 100, 1000, \cdots \tag{16}$$

To translate the take-off position information and departure sequence priority information in the encoding into a specific sortie-scheduling scheme, decoding was required. Definition: set Wg to be scheduled; set Sg already scheduled. We first extracted the integer part of the code, to obtain the take-off position information of the fleet. We then extracted the fractional part of the code, to obtain the take-off sequence in order of priority. SL ws the set of completed operations of the carrier aircraft, where SL_i was the number of operations completed by aircraft i. The decoding process for mixed fleet sortie scheduling was as follows:

Step 1 Initialization $W_g = \boldsymbol{\pi}$, $S_g = \varnothing$, $\boldsymbol{SL} = [0]_{1\times Nf}$, $index = 1$.

Step 2 When $W_g = \boldsymbol{\pi}$, go to step 6.

Step 3 Select the carrier aircraft number $i^* = W_g(index)$, which corresponds to the take-off position $c^* = y_{i^*}$. If $c^* \leq 3$ meant that the carrier aircraft was a fixed-wing carrier aircraft, we executed the fixed-wing carrier aircraft sortie operations **Case 1**~**Case 4**.

According to the completion of the operations of this carrier aircraft, we selected the next operation to execute, $Switch(SL_i + 1)$.

Case 1: Fixed-wing carrier aircraft warm-up and self-check operations. In this operation, we needed to determine whether the current parking position of the i^* carrier aircraft was warmable and, if so, we executed the operation directly, and went to Case 2, $SL_{i^*} = 1$; otherwise, we needed to transfer the i^* carrier aircraft to a warmable parking position first. Before transferring, we checked if there were free warmable parking positions and, if so, we selected the one with the minimum combined transfer time from the current parking position to that position and then to the take-off position, and we then went to Case 2, $SL_{i^*} = 1$. If there were no available warmable parking positions, we checked the earliest time when the carrier aircraft had sortied its warm-up position, and recorded it. When the aircraft had left the warm-up position, we could transfer the i^* aircraft into position and go to Case 2, $SL_{i^*} = 1$; otherwise, if the i^* carrier aircraft could not complete the warm-up, we made $index = index + 1$, and went to step 2.

Case 2: Fixed-wing carrier aircraft transfer to take-off or waiting positions. Before transferring, we checked whether there was an available path. We construct a set of parking positions $\Phi_{\mathrm{p}}(P_{i^*1}, \mathrm{c}^*)$ that interfered with the path from the warm-up position P_{i^*1} to the take-off position c^*. If $\Phi_{\mathrm{p}}(P_{i^*1}, \mathrm{c}^*)$ was not an empty set, the carrier aircraft could not execute this operation, making $index = index + 1$, and proceeding to Step 2. If $\Phi_{\mathrm{p}}(P_{i^*1}, \mathrm{c}^*)$ was an empty set, we checked the earliest passable time t_1 of the path; similarly, we constructed a set of sortie runways $\Phi_{\mathrm{r}}(P_{i^*1}, \mathrm{c}^*)$ that interfered with the path from the warm-up position P_{i^*1} to the take-off position c^*, and found the earliest time t_2 that the departure runway was free, because t_1; t_2 was the earliest transfer time; the actual transfer start time t_3 was determined by the collision avoidance delay for the carrier aircraft during transfer, based on the space resource constraint referred to in Section 4.2.4. If there were no aircraft ready for take-off at the c^* take-off position, we directly transferred the i^* carrier aircraft to the take-off position, and the duration of waiting operations after the deflector took 0. If there were aircraft ready for take-off, we transferred the i^* carrier aircraft to the waiting position corresponding to the take-off position c^*, and went to Case 3, making $SL_{i^*} = 2$.

Case 3: Fixed-wing carrier aircraft ready for take-off. We applied this operation only to the carrier aircraft in the waiting position. After the carrier aircraft in its corresponding take-off position c^* had sortied, and the plate had been reset and cooled after Δtb, we transferred the carrier aircraft into the take-off position, to prepare for take-off and, making $SL_{i^*} = 3$, went to Case 4.

Case 4: Fixed-wing carrier aircraft sorties. After the previous take-off fixed-wing aircraft on the deck disappeared. or the previous take-off carrier helicopter had taken off to a safe distance, we used that time t_1 as the earliest available take-off time for the carrier aircraft, and used t_1 as the starting time to check the time t_2 when the interference parking position on the corresponding take-off runway was cleared; therefore, t_2 was the actual take-off time for the carrier aircraft; we made $SL_{i^*} = 4$, $W_g = W_g - \{i^*\}$, $S_g = S_g \cup \{i^*\}$, $index = 1$, and we then went to step 2.

If $\mathrm{c}^* \geq 4$ meant that the carrier aircraft was a carrier helicopter, we executed the carrier helicopter sortie operations **Case 2**, **Case 4**. According to the completion of the operations of this carrier helicopter, we selected the next operation to execute: $Switch(SL_i + 2)$.

Case 2: Carrier helicopter transfer to take-off position. Before transferring, we checked if there was an available path. We constructed a set of parking positions $\Phi_{\mathrm{p}}(P_{i^*1}, \mathrm{c}^*)$ that interfered with the path from the warm-up position P_{i^*1} to the take-off position c^*: if $\Phi_{\mathrm{p}}(P_{i^*1}, \mathrm{c}^*)$ was not an empty set, the carrier aircraft could not execute this operation; we made $index = index + 1$, and went to step 2; if $\Phi_{\mathrm{p}}(P_{i^*1}, \mathrm{c}^*)$ was an empty set, we checked the earliest passable time t_1 of the path. Similarly, we constructed a set of sortie runways $\Phi_{\mathrm{r}}(P_{i^*1}, \mathrm{c}^*)$ that interfered with the path from the warm-up position P_{i^*1} to the take-off position c^*, and found the earliest time t_2 that the departure runway was free, because t_1; t_2 was the earliest transfer time, and the actual transfer start time t_3 was determined by

the collision avoidance delay for the carrier aircraft during transfer, based on the space resource constraint referred to in Section 4.2.4. If there were no aircraft ready for take-off at the c* take-off position, we directly transferred the i^* carrier aircraft to the take-off position, and the duration of waiting operations after the deflector took 0. If there were aircraft ready for take-off, we transferred the i^* carrier aircraft to the waiting position corresponding to the take-off position c*, and went to Case 4, making $SL_{i^*} = 2$.

Case 4: Carrier helicopter sorties. Preparation of the carrier helicopter for take-off, such as the deployment of the rotor blades, would start immediately after transfer to the take-off position, and would be completed in t_1. After the previous take-off fixed-wing aircraft on the deck disappeared or the previous take-off carrier helicopter had taken off to a safe distance, we used that time t_2 as the earliest available take-off time for the carrier aircraft, and we used t_2 as the starting time to check t_3, when the parking position that interfered with the departure position was cleared; therefore, t_3 was the actual take-off time for the carrier aircraft, and we made $SL_{i^*} = 4$, $W_g = W_g - \{i^*\}$, $S_g = S_g \cup \{i^*\}$, $index = 1$, and proceeded to Step 2.

Step 4 After the sortie, we output the sortie scenario, and calculated the objective function values as:

$$f = C_{\max} + \omega_{\mathrm{dt}}^{\mathrm{f}} \cdot TT, \tag{17}$$

where $\omega_{\mathrm{dt}}^{\mathrm{f}} \left(0 < \omega_{\mathrm{dt}}^{\mathrm{f}} \ll 1\right)$ was a weighting factor for aircraft transfer time.

Step 5 The start time of warm-up operations for a fixed-wing carrier aircraft must be later: this ensures that transfer to the take-off position is carried out immediately after the carrier aircraft has finished warming up, avoiding excess engine standby time and saving fuel.

5.3. Learning Factor

We designed the learning factors to adjust the individual learning dimensions; moreover, we defined the learning dimension α as an integer of $[1, [s' \cdot D'_{\max} + 1]]$, where s' was the learning factor that took values in the range (0, 1), and $D'_{\max}$ was the dimension of $\vec{Z}$ that represented the number of carrier aircraft. α determined the value of participation F in the subsequent discretization design of this study, as well as the maximum dimension of the effect of some operations on individuals in the population restart. α was used to balance the local search with the global search capability: the larger its value, the greater the global search capability of the algorithm.

α also determined the maximum individual dimension that this algorithm could operate for encircling prey, searching for prey, and the field forward and reverse order in the population restart: according to this feature, α gradually increased from 1 to $[s' \cdot D'_{\max} + 1]$ as the number of iterations increased. To balance the global and local search capabilities of the algorithm effectively, we used the exponential function for incrementing α as:

$$\alpha = \left[\left[s' \cdot D'_{\max} + 1 \right] \cdot \frac{t^2}{T^2} \right] + 1, \tag{18}$$

where t denoted the current number of iterations, T denoted the maximum number of iterations, and $[\bullet]$ denoted the rounding operation.

5.4. Discrete Design

We considered the discrete nature of the ASPMF, and the fact that the standard WOA algorithm was not applicable to this problem; therefore, we designed the whale optimization algorithm in a discrete manner. We adopted a differential information collaboration mechanism, inspired by [46]. For two random individuals $\vec{Z}_{r1}$ and $\vec{Z}_{r2}$ in the population, we extracted their corresponding take-off priority sequences $\vec{X}_{r1}$ and $\vec{X}_{r2}$, and used the differential information of the two with the participation degree F to obtain the transi-

tion individual priority sequence $\vec{X}_{new}$. Then, we reorganized with the original take-off information $\vec{Y}_{r1}$, to obtain new individuals $\vec{Z}_{new}$, as follows:

$$\vec{Z}_{new} = \vec{Z}_{r1} \oplus F \otimes \left(\vec{X}_{r2} - \vec{X}_{r1} \right). \tag{19}$$

The above equation contains the following two primary operations:

Step 1 We obtained information on the difference between individuals Z_{r1} and Z_{r2} at the participation level F:

$$\Delta_i = F \otimes \left(\vec{X}_{r2} - \vec{X}_{r1} \right) \tag{20}$$

For each dimension, the rationale was as follows:

$$\Delta_{i,j} = \begin{cases} \vec{X}_{r2,j} - \vec{X}_{r1,j} & rand(\cdot) < F \\ 0 & \text{other} \end{cases}, \tag{21}$$

where $rand(\cdot)$ was a random number uniformly distributed on (0, 1), and $F = \frac{\alpha}{D'_{\max}}$.

Step 2 We used differential information to update the individual's priority sequence:

$$\vec{Z}_{new} = \vec{Z}_{r1} \oplus \Delta_i \tag{22}$$

For each dimension, the rationale was as follows:

$$\vec{X}'_{r1,\vec{S}(j)} = \vec{X}_{r1,j}, \tag{23}$$

where $\vec{S}(j)$ was the priority designation of the carrier aircraft originally located at position j after the individual $\vec{Z}_{r1}$ had moved, using differential information Δ_i.

For example, for individual $\vec{Z}_{r1} = \left(z_1, z_2, z_3, \cdots, z_{Nf} \right)$, the subscript vector of its corresponding position was $\vec{S}_{r1} = (1, 2, 3, \cdots, Nf)$, and the subscript vector after shifting using the difference information was $\vec{S} = \vec{S}_{r1} + \Delta_i$, sorting the subscripts in $\vec{S}$ in ascending order, while the $\vec{X}_{r1}$ extracted from $\vec{Z}_{r1}$ was synchronously rearranged according to the $\vec{S}$ subscript exchange sequence, to obtain $\vec{X}'_{r1}$, and $\vec{X}'_{r1}$ was regrouped with $\vec{Y}_{r1}$ to obtain $\vec{Z}_{new}$. If the same subscript appeared in the sorting process $\vec{S}$, the subscript before sorting to the left was sorted to the right, to ensure the feasibility and diversity of the operation. If there were identical subscripts in the sorting process $\vec{S}$, the subscripts to the left before sorting were sorted to the right, to ensure the feasibility and diversity of the operation. The basic process is illustrated in Figure 6.

5.4.1. New Encircling Prey

For the current individual $\vec{Z}(t)$ and the population optimum individual $\vec{Z}_p(t)$:

$$\vec{Z}(t+1) = \vec{Z}(t) \oplus F \otimes \left(\vec{X}(t) - \vec{X}_p(t) \right). \tag{24}$$

5.4.2. New Bubble-Net Attacking Method

For the current individual $\vec{Z}(t)$ and the population optimum individual $\vec{Z}_p(t)$:

$$\vec{Z}(t+1) = \vec{Z}(t) \oplus F' \otimes \left(\vec{X}(t) - \vec{X}_p(t) \right), \tag{25}$$

where $F' = \left|\left[D'_{\max} \cdot e^{bl} \cdot \cos(2\pi l)\right]\right|$, $D'_{\max}$ was the maximum individual dimension, b was a constant that defined the shape of the spiral, l was a random number in $(-1,1)$, $[\cdot]$ was rounding, and $|\cdot|$ was the absolute value.

5.4.3. New Search for Prey

For current individuals $\overrightarrow{Z}(t)$ and random individuals $\overrightarrow{Z}_r(t)$ in the population:

$$\overrightarrow{Z}(t+1) = \overrightarrow{Z}(t) \oplus F \otimes \left(\overrightarrow{X}(t) - \overrightarrow{X}_r(t)\right). \tag{26}$$

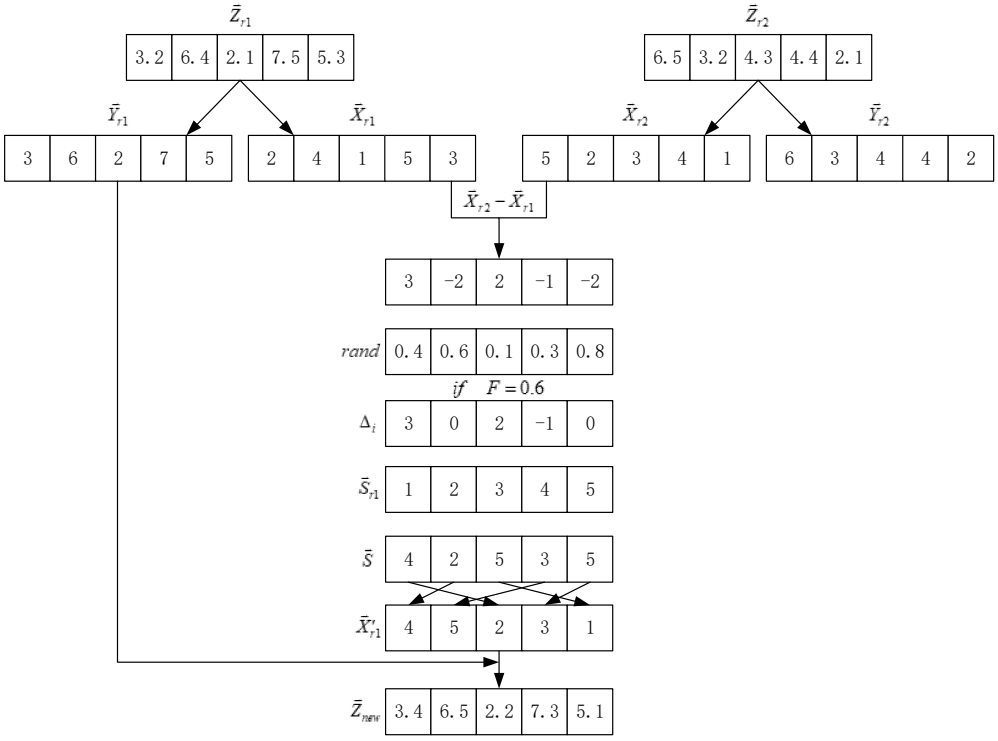


Figure 6. Diagram of differential information collaboration.

5.5. Pre-Constrained

We considered that there was interference between the parking position and take-off position in the process of carrier aircraft sortie, so when the algorithm generated a new individual, according to the analysis of the transit scheduling decoding, the individual sequence could be pre-searched for pre-adjustment of the carrier aircraft sortie sequence. For example, when an interfering aircraft position corresponding to the take-off position of carrier A existed for carrier B, and the take-off sequence of A was superior to that of B, we moved A to the latter priority of B, to further refine the effective search space.

5.6. Parameter Improvements

In the WOA framework, the algorithm encircles the prey when $|A| \leq 1$, and searches for prey when $|A| > 1$: these two foraging methods are the concrete embodiment of local search and global search, and the convergence factor $\overrightarrow{a}$ controls the value of $\overrightarrow{A}$; therefore, the change in $\overrightarrow{a}$ controls the balance between the algorithm's global and local search abilities. We improved the decreasing method of the convergence factor $\overrightarrow{a}$ by using an exponential function during the iteration of the algorithm. Through this operation, the

convergence factor showed a nonlinear dynamic change, which could effectively balance the global search and local search according to different stages:

$$\vec{a} = 2\left(1 - \frac{t^2}{T^2}\right) \tag{27}$$

5.7. Population Restart

When the population's optimal solution had not been updated for all L generations, we introduced a population restart. In this operation, we employed multiple strategies to hierarchically reconfigure the population, and instead of operating on only $\vec{X}$ in individual $\vec{Z}$ in Section 5.4, we operated on $\vec{X}$ and $\vec{Y}$ in individual $\vec{Z}$ separately, and then formed them into new individuals. With this operation, we changed the initial array of the hybrid fleet and take-off sequence, making them more diverse, and fully accounting for the factors affecting the objective function.

Inspired by PSO, we obtained a new population, by operating on an individual's historical optimal solution and the population's historical optimal solution, as shown in Table 2, where: sizepop is the number of individuals in the population; floor is the integer function to negative infinity; $\vec{Z}_{i,p}$ is the historical optimal solution for individual i, corresponding to $\vec{X}_{i,p}$ and $\vec{Y}_{i,p}$; $\vec{Z}_p$ is the population optimal solution, corresponding to $\vec{X}_p$ and $\vec{Y}_p$; $\vec{Z}_{i,new}$ is the recombination of individuals for individual i; 06 and 09 are field reinsertion operations; 12 and 15 are two-point crossover operations; and 18 and 21 are field inverse order operations.

There was an optimal sequence of sorties for each initial layout; therefore, there was a two-level search for optimality: one was used to optimize the initial layout, and the other optimized the outgoing sequence for certain layout conditions. Through this restart mechanism, the algorithm exhaustively searched the solution space.

Table 2. Population restart.

Function: Population Restart	
01:	**if** the population optimal solution is not updated for L iterations;
02:	**for** i = 1: floor(sizepop*0.4)
03:	Randomly generate a new individual to replace pop(i)
04:	**end for**
05:	**for** i = floor(sizepop*0.4) + 1: floor(sizepop*0.5)
06:	Perform a field reinsertion operation on $\vec{X}_{i,p}$ and $\vec{Y}_{i,p}$ to get $\vec{Z}_{i,new}$;
07:	**end for**
08:	**for** i = floor(sizepop*0.5) + 1: floor(sizepop*0.6)
09:	Perform a field reinsertion operation on $\vec{X}_p$ and $\vec{Y}_p$ to get $\vec{Z}_{i,new}$;
10:	**end for**
11:	**for** i = floor(sizepop*0.6) + 1: floor(sizepop*0.7)
12:	Perform a two-point crossover operation on $\vec{X}_{i,p}$ and $\vec{Y}_{i,p}$ to get $\vec{Z}_{i,new}$;
13:	**end for**
14:	**for** i = floor(sizepop*0.7) + 1: floor(sizepop*0.8)
15:	Perform a two-point crossover operation on $\vec{X}_p$ and $\vec{Y}_p$ to get $\vec{Z}_{i,new}$;
16:	**end for**
17:	**for** i = floor(sizepop*0.8) + 1: floor(sizepop*0.9)
18:	Perform a field reverse operation on $\vec{X}_{i,p}$ and $\vec{Y}_{i,p}$ to get $\vec{Z}_{i,new}$;
19:	**end for**
20:	**for** i = floor(sizepop*0.9) + 1: sizepop
21:	Perform a field reverse operation on $\vec{X}_p$ and $\vec{Y}_p$ to get $\vec{Z}_{i,new}$;
22:	**end for**
23:	**end if**

1. Field reinsertion operations

For each individual, we randomly selected any field up to α in length at any position, extracted it, and reinserted it at any position in the remaining fields, as shown in Figure 7.

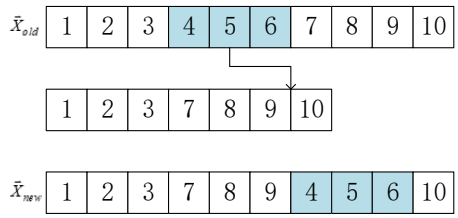


Figure 7. Field reinsertion operations.

2. Two-point crossover operations

For each individual, we randomly selected two different positions in the field, and swapped the data from those two positions, as shown in Figure 8:

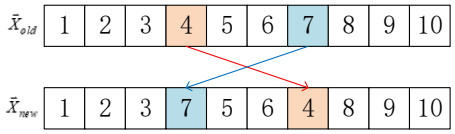


Figure 8. Two-point crossover operations.

3. Field inverse order operations

For each individual, we randomly selected fields of length no greater than α at any position, and inverted the order of the selection sequence, as shown in Figure 9:

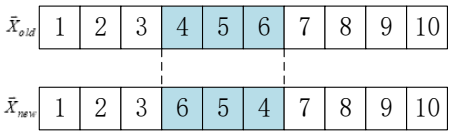


Figure 9. Field inverse order operations.

5.8. IWOA Processes

We propose a discretized improved whale algorithm for the ASPMF problem, which is divided into four main steps: (1) population initialization, (2) pre-constrained retrieval, (3) encircling prey and searching for prey, and (4) population restart. Its flow is shown in Figure 10, where p is a constant of [0, 1], l the population's optimal solution-updated generation, L the population restart threshold, t the current, and T the maximum evaluation generation of the algorithm:

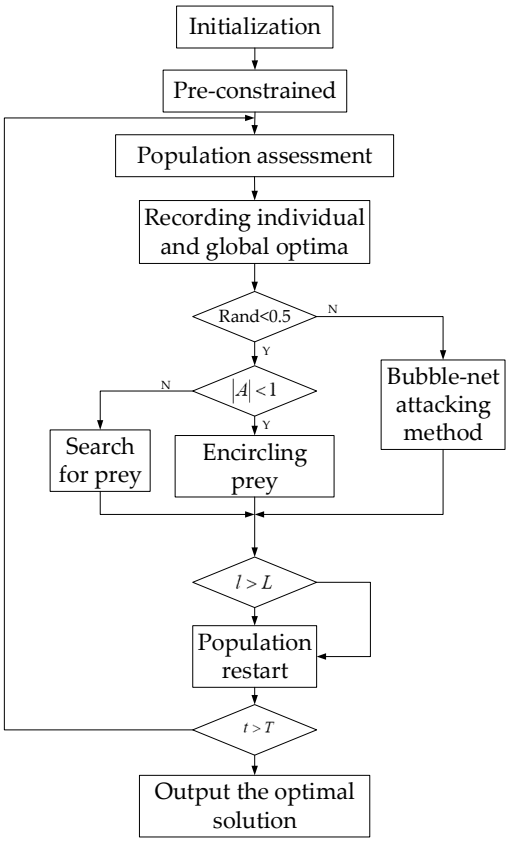


Figure 10. Flow chart of IWOA.

6. Case Study

To validate the applicability of our model and algorithm to the ASPMF, we designed three cases of 8, 12, and 16 aircraft sorties.

In the 8-aircraft sortie case, the set of positions was $\Omega_8 = [A1, A6, A7, A8, A9, A10, A11, A15]$, because any position could be used to warm up the fixed-wing aircraft, and was not an interfering position in the take-off position. In the 12-aircraft sortie case, the set of positions was $\Omega_{12} = [A1, A6, A7, A8, A9, A10, A11, A12, A13, A14, A15, A16]$, because any position could be used to warm up the fixed-wing aircraft. In the 16-aircraft sortie case, the set of positions contained all the A-positions.

6.1. Orthogonal Optimization of Algorithm Parameters

In the ASPMF case study, we defined the tethering/untethering time as 30 s, and determined that the transfer speed of the aircraft should not exceed 3.6 km/h, according to the safety requirements of surface transfer. For the algorithm parameter tuning, we used $L_{16}(3^4)$ orthogonal tables as shown in Table 3 to find the optimal values for the four parameters: population size $Np = \{10, 50, 100\}$, learning factor $s' = \{0.2, 0.6, 1\}$, population restart threshold $L = \{5, 10, 20\}$, and foraging mode selection threshold $p = \{0.3, 0.5, 0.8\}$; each parameter had four levels.

We conducted a case study for the aforementioned nine parameter combinations, each simulated in a 16-aircraft mission containing 4 carrier helicopters and 12 fixed-wing carrier aircraft. We conducted 20 experiments for each combination of parameters, and took the average transit completion times for analysis. The program was run in MATLAB 2020a with an Intel(R) Core(TM) i7-10750H CPU @ 2.60 GHz 2.59 GHz, and the experimental results as shown in Table 4:

Table 3. Orthogonal table of algorithm parameters.

	Np	***s′***	***p***	***L***
1	10	0.2	0.3	5
2	10	0.6	0.5	10
3	10	1.0	0.8	20
4	50	0.2	0.5	20
5	50	0.6	0.8	5
6	50	1.0	0.3	10
7	100	0.2	0.8	10
8	100	0.6	0.3	20
9	100	1.0	0.5	5

Table 4. Orthogonal table of algorithm parameters.

	Np	***s′***	***p***	***L***	C_{max} **(min)**
1	10	0.2	0.3	5	20.350118
2	10	0.6	0.5	10	19.836323
3	10	1.0	0.8	20	20.645739
4	50	0.2	0.5	20	20.492237
5	50	0.6	0.8	5	20.017508
6	50	1.0	0.3	10	19.644733
7	100	0.2	0.8	10	19.756210
8	100	0.6	0.3	20	20.416650
9	100	1.0	0.5	5	19.679947
O_{1j}	60.832180	60.598565	60.411501	60.047573	
O_{2j}	60.154478	60.270482	60.008508	59.237267	
O_{3j}	59.852808	59.970419	60.419457	61.554626	
R_j	0.9793724	0.6281466	0.4109494	2.3173595	
Parameter priority	*L—Np—s′—p*				
Optimum solution	*Np = 100, s′ = 1.0, p = 0.5, L = 10*				

In the table, O_{1j}, O_{2j}, and O_{3j} denote the sum of C_{max} corresponding to the *j*th *Parameter* at levels 1, 2, and 3, respectively:

$$\begin{array}{rcl} O_{11} & = & C_{max1} + C_{max2} + C_{max3} \\ O_{14} & = & C_{max1} + C_{max5} + C_{max9} \end{array} \quad (28)$$

R_j denotes the extreme differences among O_{1j}, O_{2j}, and O_{3j} for the same parameter. The larger R_j was, the more pronounced was the effect of the *Parameter* on C_{max}. The optimal levels of O_{1j}, O_{2j}, and O_{3j} were those with the smallest C_{max}. From the data in the table, the optimal solution was: population size $Np = 100$, learning factor $s' = 1$, foraging mode selection threshold $p = 0.5$, and population restart threshold $L = 10$.

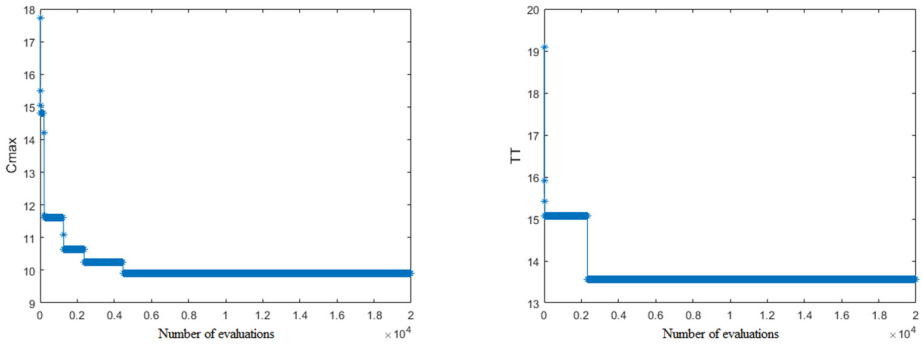
6.2. Simulation of Different Sortie Cases

We selected the optimal solution for the above algorithm parameters. We used a decoding evaluation number of 20,000 as the algorithm iteration termination condition, to simulate and optimize the three cases.

6.2.1. 8-Aircraft Sortie Case

After simulation and optimization, we obtained the trend of the optimization objective of the sortie scheduling, as shown in Figure 11. The sortie completion time converged to the optimal value of C_{max} = 9.91 min, and the fleet transfer time TT = 13.57 min. From Figure 11a,b, it can be seen that the sortie completion time gradually converged to the

optimal solution when the number of evaluations reached approximately 4500. In contrast, the fleet transfer time converged to the optimal solution when the number of evaluations reached 2000. The fixed-wing aircraft could all finish warming up in their original position; therefore, when the deck arrangement of the aircraft and their take-off positions were determined, the transit time was also fixed. In the optimization process, the first step was to complete the optimization of the fleet layout and take-off slot allocation, at which point the TT had reached the optimal solution. Subsequently, the optimal sequence of sorties was determined based on the optimal layout, at which point the sortie completion time converged; hence, the convergence of (b) before (a).



(**a**) (**b**)

Figure 11. Trends in scheduling optimization objectives in 8-aircraft sortie case: (**a**) trends in fleet sortie completion time in 8-aircraft sortie case; (**b**) trends in fleet transit time in 8-aircraft sortie case.

The same conclusion can be drawn by looking at the trend of the optimization indices for the 12-aircraft sortie.

Figures 12 and 13 show the sortie Gantt chart and the Gantt chart for each take-off position for the optimal mixed fleet sortie solution. In Figure 13, each line of the Gantt bars represents a carrier aircraft, with the number shown as the vertical coordinate, and time as the horizontal coordinate; "$J-j$" in the Gantt box represents the departure operation j, which is not shown when the second operation of the fixed-wing carrier aircraft took very little time. The transit process of the carrier aircraft is shown in Figure 13 in gray gantry boxes, with the start and target parking positions marked at the left and right ends of the gray gantry boxes. In Figure 14, each row of the Gantt bars represents the take-off position, and is numbered as shown in the vertical coordinates. The numbers in the Gantt box are labeled as "Carrier Aircraft Number-Sortie Operation Stage Number".

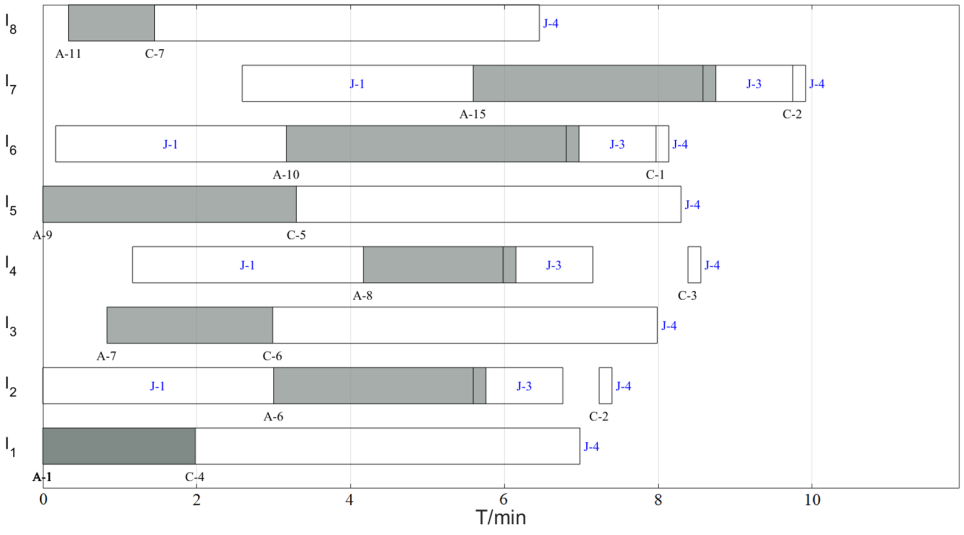


Figure 12. Gantt chart of sortie schedule in 8-aircraft sortie case.

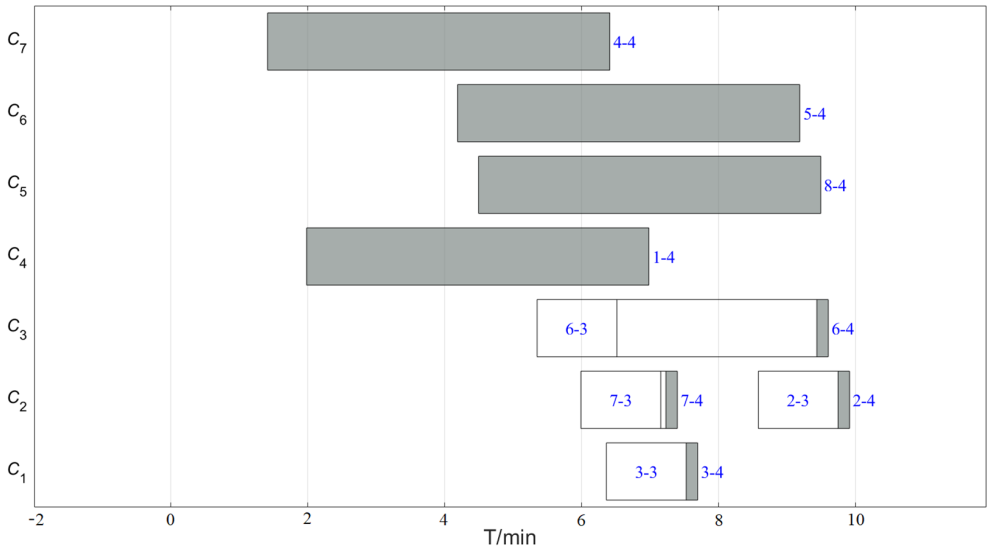


Figure 13. Gantt chart of take-off position occupancy in 8-aircraft sortie case.

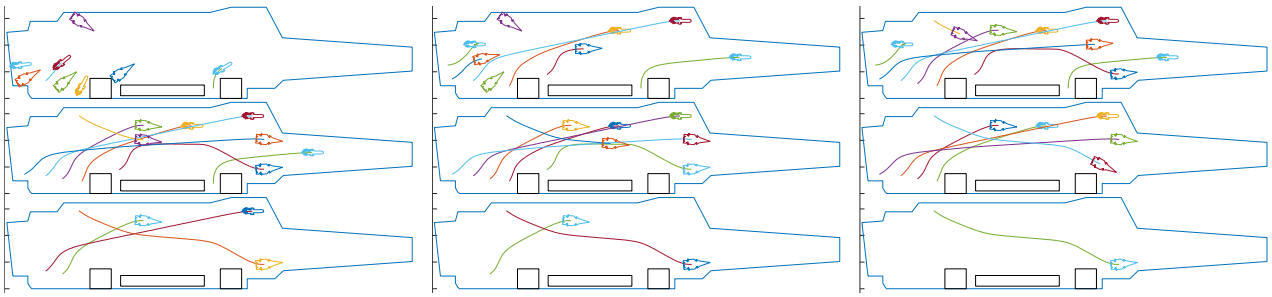


Figure 14. Diagram of the transfer process in 8-aircraft sortie case.

These two Gantt charts demonstrate that the sortie scheduling process satisfied the model constraints. From Figure 13, I2's take-off position C2 was interfered with by C4. Although I2 had completed its pre-take-off preparation before I1 took off, it still needed to wait for I1, which occupied the C4 take-off position; I3, I4, I5, and I6 also satisfied this constraint, and they all verified the validity of the model and algorithm. By combining Figures 12–14, we can see that the collision avoidance constraint was satisfied by the aircraft fleet.

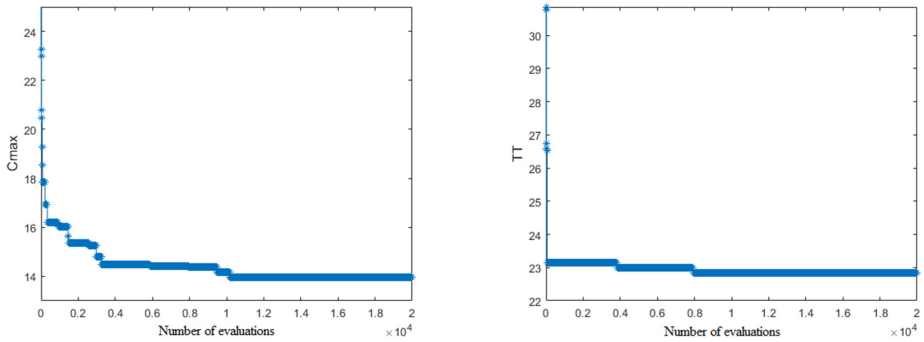
6.2.2. 12-Aircraft Sortie Case

After simulation and optimization, we obtained the trend of the optimization objective of the sortie scheduling, as shown in Figure 15. The sortie completion time converged to the optimal value of C_{max} = 13.96 min, and the fleet transfer time TT = 22.85 min. Figures 16–18 show the relevant images of the optimal solution for this case.

Comparing the images of the 12-aircraft with the 8-aircraft sortie, we can see that one feature of the 12-aircraft sortie was that no helicopters took off from the C4 take-off position: this was because, in the 12-aircraft sortie case, the fixed-wing carrier aircraft increased from four to eight. No helicopters took off from the C4, as it would have interfered with the C1, C2, and C3 take-off positions simultaneously, and sortieing the carrier helicopters here would have caused the fleet to take much longer to sortie.

6.2.3. 16-Aircraft Sortie Case

After the simulation and optimization, we obtained the trend of the optimization objective of the sortie scheduling, as shown in Figure 19, which converged to the optimal value of C_{max} = 19.01 min, and the fleet transfer time TT = 31.46 min. Figures 20–22 show the relevant images of the optimal solution in this case.



(**a**) (**b**)

Figure 15. Trends in scheduling optimization objectives in 12-aircraft sortie case: (**a**) trends in fleet sortie completion time in 12-aircraft sortie case; (**b**) trends in fleet transit time in 12-aircraft sortie case.

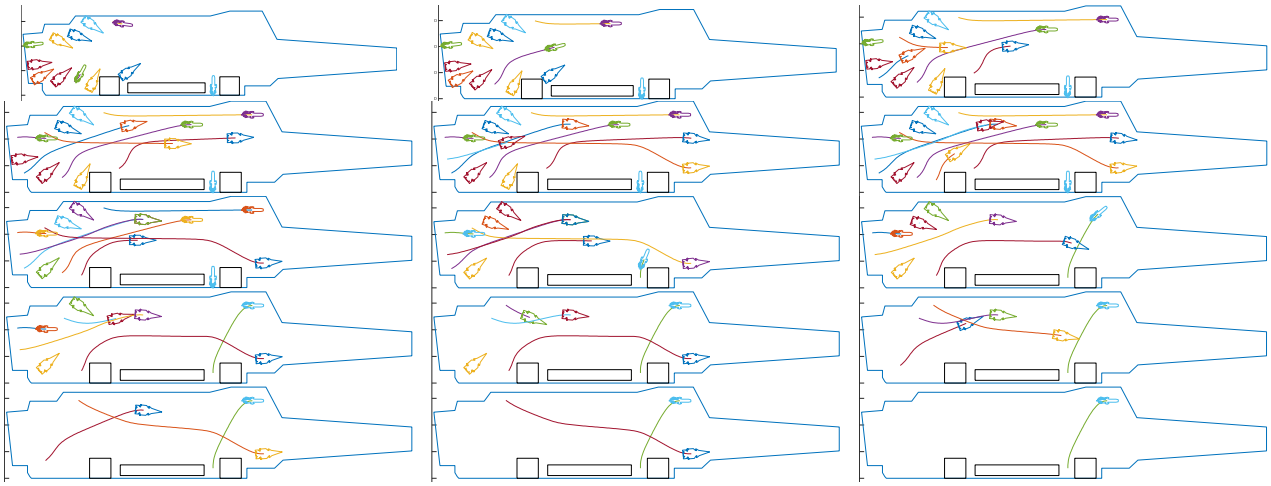


Figure 16. Diagram of the transfer process in 12-aircraft sortie case.

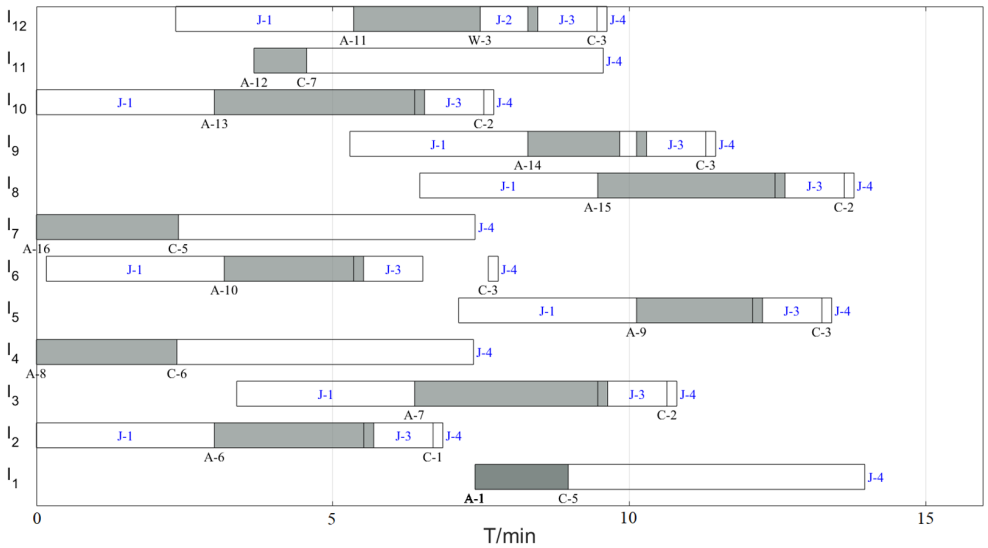


Figure 17. Gantt chart of sortie schedule in 12-aircraft sortie case.

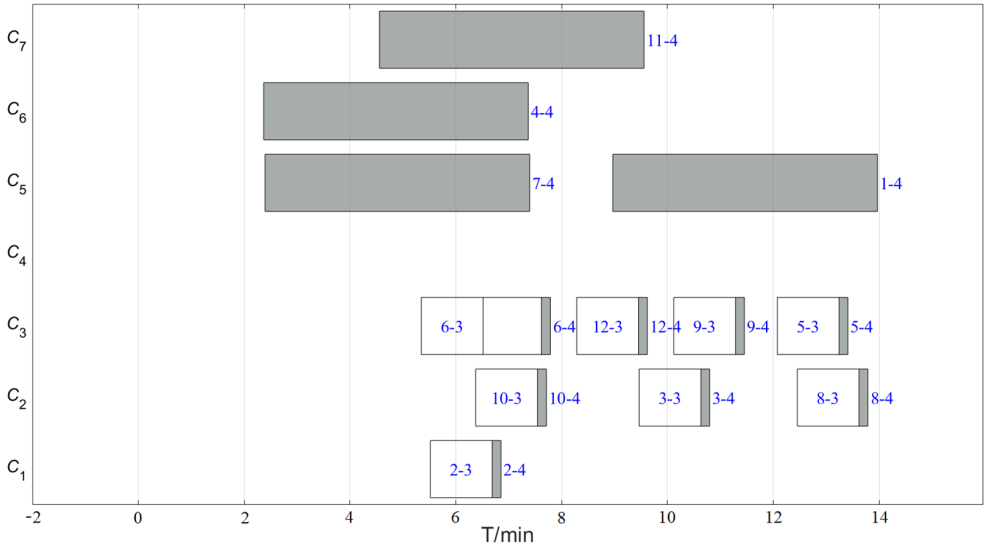
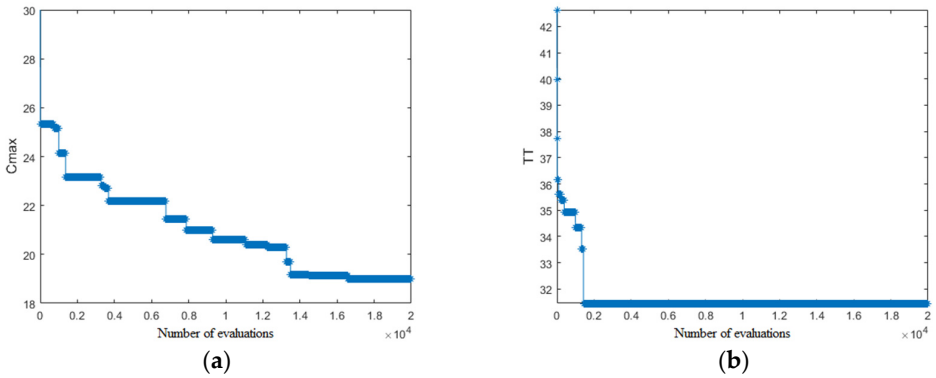


Figure 18. Gantt chart of take-off position occupancy in 12-aircraft sortie case.



(**a**) (**b**)

Figure 19. Trends in scheduling optimization objectives in 16-aircraft sortie case: (**a**) trends in fleet sortie completion time in 16-aircraft sortie case; (**b**) trends in fleet transit time in 16-aircraft sortie case.

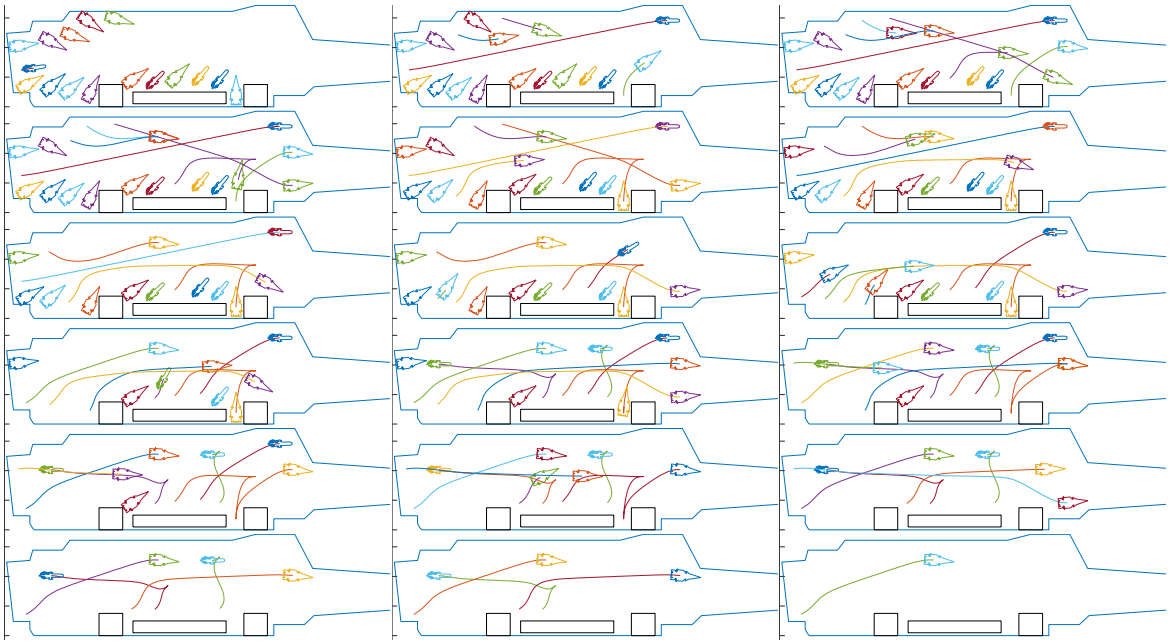


Figure 20. Diagram of the transfer process in 16-aircraft sortie case.

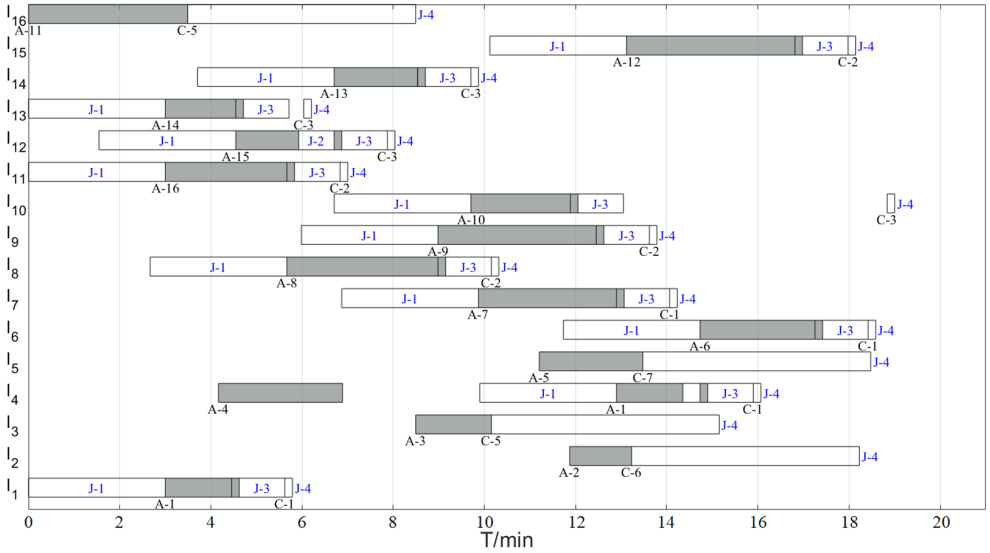


Figure 21. Gantt chart of sortie schedule in 16-aircraft sortie case.

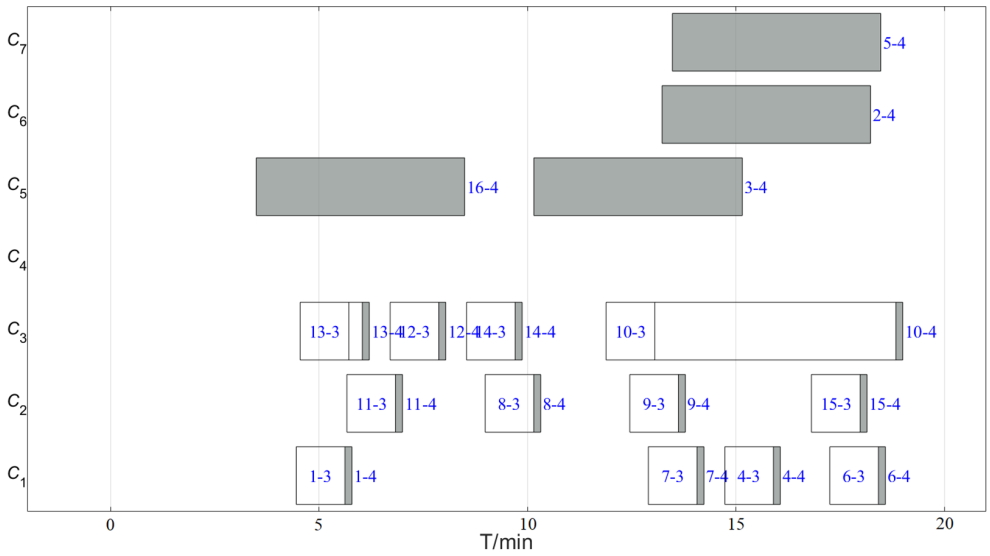


Figure 22. Gantt chart of take-off position occupancy in 16-aircraft sortie case.

6.3. Comparison of Different Sortie Strategies

To verify the superiority of our proposed sortie strategy, we compared it with the current carrier sortie strategy (carrier helicopters first, followed by fixed-wing carrier aircraft). We chose the optimal parameter scheme of the previous section, and the algorithm iteration termination condition was taken as the number of evaluations in 20,000. We carried out simulation experiments of the two departure strategies for the three cases. The results obtained are shown in Table 5:

Table 5. Simulation results for different sortie strategies.

Case	Strategies	C_{max} (min)	Result
8-aircraft sortie	Hybrid sortie	9.91	0.02 min reduction
	Separate sortie	9.93	
12-aircraft sortie	Hybrid sortie	13.96	0.10 min reduction
	Separate sortie	14.06	
16-aircraft sortie	Hybrid sortie	19.01	1.32 min reduction
	Separate sortie	20.33	

According to Table 5, the difference between the 8- and 12-aircraft fleets was insignificant, and the time reduction was within 0.1 min for both sorties. In the sortie cases we set, the initial parking positions used in the 8- and 12-aircraft cases were warmable. The fixed-wing aircraft did not need to be transferred for the warm-up; thus, there was relatively little path interference during the fleet sortie. The transfer and operation time of the carrier aircraft were constant; therefore, the carrier aircraft sortie time was also relatively constant (for the optimal solution only). The time difference between the two sortie strategies was mainly due to the constraint that the fixed-wing carrier aircraft had to take off later than the carrier helicopter; therefore, the time difference between the two sortie strategies arose mainly from the mandatory waiting time for a fixed-wing carrier aircraft.

The hybrid departure strategy in the 16-engine departure case reduced by 1.32 min, due to the complex warm-up transfer involved in the fixed-wing carrier aircraft in this case, and to a large number of aircraft in the fleet, which used more space. The change in the sortie sequence due to the mandatory requirement of the sortie strategy led to a change in the interference between the warm-up and take-off transfer paths, resulting in delayed sorties, to avoid collisions. The increased use of the C4 take-off position owing to the separate sortie strategy, which blocked the sorties of all fixed-wing aircraft, also contributed to the extended sortie times of the fleet.

7. Conclusions

To solve the ASPMF problem, we propose a method that can integrate the kinematic constraints of carrier aircraft, the terminal attitude constraints of carrier aircraft, the deck space resource constraints, the exclusivity constraints of equipment allocation, the timing constraints of carrier aircraft transfer, the collision avoidance constraints of carrier aircraft transfer, and the timing constraints of transferring equipment. This method includes two parts: path library construction and ASPMF optimization modeling.

In the path library construction part, we used a combination of heuristic and optimal control deck path planning algorithms to carry out single aircraft path planning for carrier aircraft scheduling in a full-layout state; we obtained the shortest path for carrier aircraft transfer that satisfied the carrier aircraft kinematics, terminal attitude constraints, and space resource constraints; and we successfully built the ASPMF path library.

In the model building part, we modeled the ASPMF with reference to the HFSP. The minimization of the sortie time and transit time of the fleet was the objective of optimization; simultaneously, we considered the constraints of the process flow, space resources, and support resources in the sortie process.

To find the optimal solution of the model efficiently and stably, we designed the IWOA, using the WOA as a framework. We discretized the algorithm to make it suitable for the ASPMF problem, improved it using learning factors, pre-constraints, parameter improvements, and population restarts, and used $L_{16}(3^4)$ orthogonal tables to determine the optimal combination of parameters for the algorithm.

Finally, in the case study, three cases of 8, 12, and 16 aircraft sorties were constructed, each with four carrier helicopters. We used these three cases to verify the model's validity and the performance of the IWOA optimization, and then compared our proposed hybrid sortie strategy with the existing separate sortie strategy: the results show that the hybrid sortie strategy can effectively improve the efficiency of fleet sorties, and is more effective for large-scale sorties.

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

Funding: This research work was supported by the National Natural Science Foundation of China (grant number 52102453) and Chongqing Research Program of Basic Research and Frontier Technology (grant number cstc2020jcyj-msxmX0602).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

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

Conflicts of Interest: There is no conflict of interest.

References

1. Michini, B.; How, J. A Human-Interactive Course of Action Planner for Aircraft Carrier Deck Operations. In Proceedings of the Infotech@Aerospace 2011, St. Louis, MO, USA, 29–31 March 2011; American Institute of Aeronautics and Astronautics: St. Louis, MO, USA, 2011.
2. Ryan, J.; Cummings, M.; Roy, N.; Banerjee, A.; Schulte, A. Designing an Interactive Local and Global Decision Support System for Aircraft Carrier Deck Scheduling. In Proceedings of the Infotech@Aerospace 2011, St. Louis, MO, USA, 29–31 March 2011; American Institute of Aeronautics and Astronautics: St. Louis, MO, USA, 2011.
3. Ryan, J.C.; Banerjee, A.G.; Cummings, M.L.; Roy, N. Comparing the Performance of Expert User Heuristics and an Integer Linear Program in Aircraft Carrier Deck Operations. *IEEE Trans. Cybern.* **2014**, *44*, 761–773. [CrossRef] [PubMed]
4. Ryan, J.C.; Cummings, M.L. Assessing the Performance of Human-Automation Collaborative Planning Systems. Ph.D. Thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, 2011.
5. Cui, R.; Han, W.; Su, X.; Zhang, Y.; Guo, F. A multi-objective hyper heuristic framework for integrated optimization of carrier-based aircraft flight deck operations scheduling and resource configuration. *Aerosp. Sci. Technol.* **2020**, *107*, 106346. [CrossRef]
6. Yuan, P.; Han, W.; Su, X.; Liu, J.; Song, J. A Dynamic Scheduling Method for Carrier Aircraft Support Operation under Uncertain Conditions Based on Rolling Horizon Strategy. *Appl. Sci.* **2018**, *8*, 1546. [CrossRef]
7. Zheng, M.; Yang, F.; Dong, Z.; Xie, S.; Chu, X. Carrier-borne aircrafts aviation operation automated scheduling using multiplicative weights apprenticeship learning. *Int. J. Adv. Robot. Syst.* **2019**, *16*, 1729881419828917. [CrossRef]
8. Liu, J.; Han, W.; Peng, H.; Wang, X. Trajectory planning and tracking control for towed carrier aircraft system. *Aerosp. Sci. Technol.* **2019**, *84*, 830–838. [CrossRef]
9. Wu, Y.; Sun, L.; Qu, X. A sequencing model for a team of aircraft landing on the carrier. *Aerosp. Sci. Technol.* **2016**, *54*, 72–87. [CrossRef]
10. Wu, Y.; Wang, Y.; Qu, X.; Sun, L. Exploring mission planning method for a team of carrier aircraft launching. *Chin. J. Aeronaut.* **2019**, *32*, 1256–1267. [CrossRef]
11. Guo, F.; Han, W.; Su, X.; Liu, Y.; Cui, R. A bi-population immune algorithm for weapon transportation support scheduling problem with pickup and delivery on aircraft carrier deck. *Def. Technol.* **2021**, S2214914721002282. [CrossRef]
12. Yuan, Y.; Xu, H. Multiobjective Flexible Job Shop Scheduling Using Memetic Algorithms. *IEEE Trans. Autom. Sci. Eng.* **2015**, *12*, 336–353. [CrossRef]
13. Driss, I.; Mouss, K.N.; Laggoun, A. A new genetic algorithm for flexible job-shop scheduling problems. *J. Mech. Sci. Technol.* **2015**, *29*, 1273–1281. [CrossRef]
14. Liu, J.; Han, W.; Liu, C.; Peng, H. A New Method for the Optimal Control Problem of Path Planning for Unmanned Ground Systems. *IEEE Access* **2018**, *6*, 33251–33260. [CrossRef]
15. Han, W.; Liu, Z.; Su, X.; Cui, K.; Liu, J. Deck path planning of carrier-based aircraft based on heuristic and optimal control. *Syst. Eng. Electron.* **2022**, *1*, 1–16. Available online: https://kns.cnki.net/kcms/detail/11.2422.TN.20220507.1357.002.html (accessed on 26 October 2022).
16. Mirjalili, S.; Lewis, A. The Whale Optimization Algorithm. *Adv. Eng. Softw.* **2016**, *95*, 51–67. [CrossRef]
17. Su, X.; Han, W.; Wu, Y.; Zhang, Y.; Liu, J. A Proactive Robust Scheduling Method for Aircraft Carrier Flight Deck Operations with Stochastic Durations. *Complexity* **2018**, *2018*, 6932985. [CrossRef]
18. 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.* 2022; *in press*. [CrossRef]
19. Hu, H.; Wu, Y.; Xu, J.; Sun, Q. Path Planning for Autonomous Landing of Helicopter on the Aircraft Carrier. *Mathematics* **2018**, *6*, 178. [CrossRef]
20. Su, X.; Li, Z.; Song, J.; Wang, L. A Path Planning Method for Carrier Aircraft on Deck Combining Artificial Experience and Intelligent Search. *IOP Conf. Ser. Mater. Sci. Eng.* **2018**, *381*, 12194. [CrossRef]
21. Zhenxing, Y.; Ziyu, Y.; Wenhao, W.; Yichen, C.; Huiyuan, C. The enemy air-threat prediction based aircraft real-time path planning for offshore combat. In Proceedings of the 2018 IEEE 8th International Conference on Underwater System Technology: Theory and Applications (USYS), Wuhan, China, 1–3 December 2018; pp. 1–4.
22. Si, W.; Sun, T.; Song, C.; Zhang, J. Design and verification of a transfer path optimization method for an aircraft on the aircraft carrier flight deck. *Front. Inf. Technol. Electron. Eng.* **2021**, *22*, 1221–1233. [CrossRef]

23. Komaki, G.M.; Sheikh, S.; Malakooti, B. Flow shop scheduling problems with assembly operations: A review and new trends. *Int. J. Prod. Res.* **2019**, *57*, 2926–2955. [CrossRef]
24. Ben-Yehoshua, Y.; Mosheiov, G. A single machine scheduling problem to minimize total early work. *Comput. Oper. Res.* **2016**, *73*, 115–118. [CrossRef]
25. Yoo, J.; Lee, I.S. Parallel machine scheduling with maintenance activities. *Comput. Ind. Eng.* **2016**, *101*, 361–371. [CrossRef]
26. Çaliş, B.; Bulkan, S. A research survey: Review of AI solution strategies of job shop scheduling problem. *J. Intell. Manuf.* **2015**, *26*, 961–973. [CrossRef]
27. Shen, L.; Dauzère-Pérès, S.; Neufeld, J.S. Solving the flexible job shop scheduling problem with sequence-dependent setup times. *Eur. J. Oper. Res.* **2018**, *265*, 503–516. [CrossRef]
28. Literature Review of Open Shop Scheduling Problems. Available online: https://www.scirp.org/html/4-8701327_53605.htm (accessed on 18 June 2022).
29. Lin, S.-W.; Cheng, C.-Y.; Pourhejazy, P.; Ying, K.-C.; Lee, C.-H. New benchmark algorithm for hybrid flowshop scheduling with identical machines. *Expert Syst. Appl.* **2021**, *183*, 115422. [CrossRef]
30. Meng, L.; Zhang, C.; Shao, X.; Ren, Y.; Ren, C. Mathematical modelling and optimisation of energy-conscious hybrid flow shop scheduling problem with unrelated parallel machines. *Int. J. Prod. Res.* **2019**, *57*, 1119–1145. [CrossRef]
31. Yu, C.; Semeraro, Q.; Matta, A. A genetic algorithm for the hybrid flow shop scheduling with unrelated machines and machine eligibility. *Comput. Oper. Res.* **2018**, *100*, 211–229. [CrossRef]
32. Meng, L.; Zhang, C.; Shao, X.; Zhang, B.; Ren, Y.; Lin, W. More MILP models for hybrid flow shop scheduling problem and its extended problems. *Int. J. Prod. Res.* **2020**, *58*, 3905–3930. [CrossRef]
33. Zhang, C.; Tan, J.; Peng, K.; Gao, L.; Shen, W.; Lian, K. A discrete whale swarm algorithm for hybrid flow-shop scheduling problem with limited buffers. *Robot. Comput.-Integr. Manuf.* **2021**, *68*, 102081. [CrossRef]
34. de Siqueira, E.C.; Souza, M.J.F.; de Souza, S.R. A Multi-objective Variable Neighborhood Search algorithm for solving the Hybrid Flow Shop Problem. *Electron. Notes Discrete Math.* **2018**, *66*, 87–94. [CrossRef]
35. Kalczynski, P.J.; Kamburowski, J. On no-wait and no-idle flow shops with makespan criterion. *Eur. J. Oper. Res.* **2007**, *178*, 677–685. [CrossRef]
36. Lee, T.-S.; Loong, Y.-T. A review of scheduling problem and resolution methods in flexible flow shop. *Int. J. Ind. Eng. Comput.* **2019**, *10*, 67–88. [CrossRef]
37. Xie, J.; Gao, L.; Peng, K.; Li, X.; Li, H. Review on flexible job shop scheduling. *IET Collab. Intell. Manuf.* **2019**, *1*, 67–77. [CrossRef]
38. Wang, F.; Rao, Y.; Zhang, C.; Tang, Q.; Zhang, L. Estimation of Distribution Algorithm for Energy-Efficient Scheduling in Turning Processes. *Sustainability* **2016**, *8*, 762. [CrossRef]
39. Lu, C.; Gao, L.; Pan, Q.; Li, X.; Zheng, J. A multi-objective cellular grey wolf optimizer for hybrid flowshop scheduling problem considering noise pollution. *Appl. Soft Comput.* **2019**, *75*, 728–749. [CrossRef]
40. Dios, M.; Fernandez-Viagas, V.; Framinan, J.M. Efficient heuristics for the hybrid flow shop scheduling problem with missing operations. *Comput. Ind. Eng.* **2018**, *115*, 88–99. [CrossRef]
41. Moccellin, J.V.; Nagano, M.S.; Pitombeira Neto, A.R.; de Athayde Prata, B. Heuristic algorithms for scheduling hybrid flow shops with machine blocking and setup times. *J. Braz. Soc. Mech. Sci. Eng.* **2018**, *40*, 40. [CrossRef]
42. Chamnanlor, C.; Sethanan, K.; Gen, M.; Chien, C.-F. Embedding ant system in genetic algorithm for re-entrant hybrid flow shop scheduling problems with time window constraints. *J. Intell. Manuf.* **2017**, *28*, 1915–1931. [CrossRef]
43. Marichelvam, M.K.; Geetha, M.; Tosun, Ö. An improved particle swarm optimization algorithm to solve hybrid flowshop scheduling problems with the effect of human factors—A case study. *Comput. Oper. Res.* **2020**, *114*, 104812. [CrossRef]
44. Zhao, F.; Qin, S.; Zhang, Y.; Ma, W.; Zhang, C.; Song, H. A hybrid biogeography-based optimization with variable neighborhood search mechanism for no-wait flow shop scheduling problem. *Expert Syst. Appl.* **2019**, *126*, 321–339. [CrossRef]
45. Ren, J.; Ye, C.; Yang, F. Solving flow-shop scheduling problem with a reinforcement learning algorithm that generalizes the value function with neural network. *Alex. Eng. J.* **2021**, *60*, 2787–2800. [CrossRef]
46. Zheng, X.; Wang, L.; Wang, S. A hybrid discrete fruit fly optimization algorithm for solving permutation flow-shop sched-uling problem. *Control Theory Appl.* **2014**, *31*, 159–164. [CrossRef]