Journal of Systems Engineering and Electronics
Vol. 32, No. 2, April 2021, pp.331 – 346

A dual population multi-operator genetic algorithm for flight deck operations scheduling problem

CUI Rongwei[1], HAN Wei[1], SU Xichao[2,*], LIANG Hongyu[3], and LI Zhengyang[3]

1. Aeronautical Foundation College, Naval Aviation University, Yantai 264001, China; 2. Aeronautical Operations College, Naval Aviation University, Yantai 264001, China; 3. Unit 91404 of the PLA, Qinhuangdao 066000, China

Abstract: It is of great significance to carry out effective scheduling for the carrier-based aircraft flight deck operations. In this paper, the precedence constraints and resource constraints in flight deck operations are analyzed, then the model of the multi-aircraft integrated scheduling problem with transfer times (MAISPTT) is established. A dual population multi-operator genetic algorithm (DPMOGA) is proposed for solving the problem. In the algorithm, the dual population structure and random-key encoding modified by starting/ending time of operations are adopted, and multiple genetic operators are self-adaptively used to obtain better encodings. In order to conduct the mapping from encodings to feasible schedules, serial and parallel scheduling generation scheme-based decoding operators, each of which adopts different justified mechanisms in two separated populations, are introduced. The superiority of the DPMOGA is verified by simulation experiments.

Keywords: genetic algorithm, project scheduling, flight deck operation, transfer times of resources.

DOI: 10.23919/JSEE.2021.000028

1. Introduction

The carrier battle group is an important symbol of national maritime comprehensive strength. As the core of the offensive and defensive system of the carrier battle group, the aircraft undertakes most of the combat mission. Its sortie rate is closely related to the ability to efficiently complete flight deck operations such as inspection, fueling, and oxygen filling. The purpose of the flight deck operations scheduling problem (FDOSP) is to shorten the makespan and efficiently complete all operations by reasonably arranging resources. Compared to the ground operations of aircraft on the airport, the carrier deck space is narrower, and the performing environment is more changeable, so operation and resource constraints of the aircraft carrier are more complicated [1]. With the increasement in the sortie rate and the number of aircraft, higher requirements have been imposed on the flight deck operations scheduling capability, and the experience-based manual scheduling is inefficient in FDOSP [2].

In view of importance, complexity and the critical role in combat for FDOSP, it is of great significance to carry out effective scheduling for the aircraft fleet deck operations and resource allocation, establish a feasible scheduling model, and produce an efficient scheduling scheme [3]. In recent years, FDOSP has become an emerging hot spot and many scholars have carried out a large number of studies on this issue.

This paper focuses on the scheduling of flight deck operations for the pre-flight preparation stage. On the basis of analysis of various constraints in the pre-flight preparation stage, the multi-aircraft integrated scheduling problem with transfer times (MAISPTT) is established, and the dual population multi-operator genetic algorithm (DPMOGA) is presented for the MAISPTT. The main contributions of this research are as follows:

(i) The precedence constraints, resource constraints, and transfer times of personnel and support equipment in the pre-flight preparation stage are described in detail. The MAISPTT is formulated as a mixed integer mathematical programming model with the objective of minimizing the makespan.

(ii) A scheduling optimization algorithm called DPMOGA is proposed. The dual population structure and random-key encoding modified by starting/ending time of operations are adopted. Various types of genetic operators are proposed, and the serial scheduling generation scheme (SSGS) or the parallel scheduling generation scheme (PSGS) are used to generate feasible schedules. An adaptive selection mechanism for genetic operators is designed.

The remainder of this paper is organized as follows. Section 2 presents a brief literature review of related

Manuscript received November 20, 2020.
*Corresponding author.
This work was supported by the National Natural Science Foundation of China (61671462).

work. Section 3 describes the MAISPTT. A mathematical programming model is presented in Section 4. In Section 5, the DPMOGA is proposed. Section 6 reports and discusses the results of the simulation. Finally, this paper is concluded by Section 7.

2. Literature review

2.1 Flight deck operations for pre-flight preparation stage

There are several variations of the scheduling model for pre-flight preparation according to different degrees of abstraction, which mainly include the hybrid flow-shop scheduling model [4], the flexible job-shop scheduling model [5], and the resource-constrained project scheduling problem (RCPSP) model [6−11]. Researches based on the hybrid flow-shop scheduling model and the flexible job-shop scheduling model have disadvantages in meeting the complex networked constraints of operations, so most researches for FDOSP are based on RCPSP recently. In these researches, the support task of an aircraft is considered as a project, and the personnel and support equipment on the flight deck are considered as resources. Additionally, during the actual flight deck operations, it takes a certain amount of time to transfer personnel and support equipment from one work station to another, so the transfer times of personnel and support equipment between different work stations have to be considered.

2.2 Solving methods for RCPSPs

The RCPSP is an NP-hard combinatorial problem, and several methods for solving RCPSPs, including exact methods, heuristics and meta-heuristics, have been proposed. From the perspective of practicability and performance, the exact methods are computationally expensive for large-scale scheduling problems. Compared to exact methods, heuristics are easy to implement and computationally cheaper, but it is difficult to find an efficient priority rule that performs well for a broad range of RCPSPs [12]. The meta-heuristics use intelligent optimization algorithms to find better encoding, then generate a feasible schedule using schedule generation schemes (SGS), which provide the best trade-off between practicability and performance, and have attracted the attention of researchers and provided improvements for solving RCPSPs. Several meta-heuristics, including the differential evolution algorithm [12], improved particle swarm optimization (IPSO) [13], and the hybrid estimation of distribution algorithm (HEDA) [14], were proposed for solving the RCPSPs. As a classic evolutionary algorithm, the genetic algorithm (GA) has been widely used for RCPSPs [15−19]. To the best of our knowledge, GA has not been applied to the pre-flight preparation operations scheduling problem with resource transfer times. And how to effectively select each genetic operator to improve the search ability and convergence speed for specific problems remains to be studied.

3. Description of MAISPTT

3.1 Description of flight deck operations

The launching and landing cycle of the aircraft fleet is a multi-stage task. After the deck operations are completed, the aircraft will taxi to the designated catapult to complete launching according to the sortie plan. After completing the combat mission in the air, the aircraft finally completes the recovery mission and return to the aircraft carrier. If it is necessary to dispatch these aircraft again, they will repeat the above process [10]. Pre-flight preparation, which is the main stage of flight deck operations of FDOSP, has complicated precedence and resource constraints. After aircraft landing on the flight deck, a tractor will tow it from the temporal parking spot to the service parking spot for pre-flight preparation. Once chocked and chained, the aircraft will turn to the pre-flight preparation stage, during which inspections, alignment of inertial navigation system (INS), fueling, oxygen filling and nitrogen charging, are completed by different professional personnel. Support equipment mainly provides various types of supply resources such as fuel, ammunition, oxygen and nitrogen [20].

The launching and landing cycle of the aircraft is proceeded in a deck cycle way, which means an aircraft fleet will go through pre-flight preparation, launching and landing on the flight deck within a deck cycle. To avoid the disorder and ensure the sustainability of cyclic operations, the duration of each deck cycle is regulated, including the modes of 1+00 (1 h 0 min), 1+15 (1 h 15 min), etc. Any delay of pre-flight preparation may disturb the operating tempos under such a time-critical scheduling [11]. The goal of MAISPTT is to minimize the makespan and increase sortie rates.

3.2 Constraints of MAISPTT

Constraints of MAISPTT consist of precedence constraints and resource constraints. Specifically, resource constraints can be divided into personnel constraints, support equipment constraints, work station space constraints, and supply resource constraints.

(i) Precedence constraints

For a given sortie mission, an aircraft fleet (denoted as a set $I=\{1,2,\cdots,|I|\}$) is designated to perform the flight deck operations before launching. The operations of each aircraft $i\in I$ is denoted as $J_i=\{1,2,\cdots,|J_i|\}$, containing a

dummy starting operation, a dummy ending operation, and $|J_i|-2$ real operations. All operations of fleet are denoted as $J=\{(i,j)|i\in I, j\in J_i\}$. Referring to the RCPSP, the support task of a single aircraft can be regarded as a project, and the precedence constraints of the aircraft fleet can be described by the activity on node (AoN) network as shown in Fig. 1, where an operation is represented by a node, and the precedence constraints between activities are denoted by the arcs.

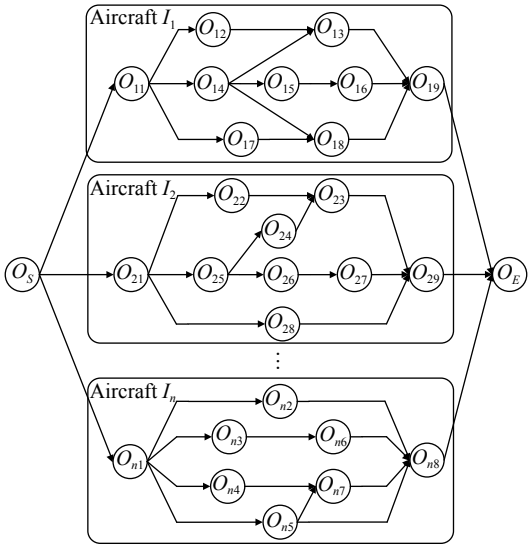


Fig. 1 AoN network for aircraft fleet

In Fig. 1, O_{ij} denotes the jth operation of the ith aircraft, O_S and O_E are the dummy starting operation and the dummy ending operation respectively. The operation O_{ij}, whose work station space is W_{ij}, cannot be performed earlier than operations in its immediate predecessors set Ψ_{ij}. The starting time and ending time of operation O_{ij} are denoted as S_{ij} and E_{ij} respectively, and $E_{ij}=S_{ij}+d_{ij}$, where d_{ij} is the duration of O_{ij}. A_{it} denotes the set of operations of aircraft i which are active in period t. Moreover, aircraft i cannot start its operations until it has been chocked and chained in the service parking spot p_i at the released time Ex_i.

(ii) Personnel constraints

Kp denotes the set of personnel trade types, Lp_k denotes the set of personnel with trade $k(k\in Kp)$, $Lp_k=\{1,2,\cdots,|Lp_k|\}$, rp_{ijk} denotes the number of personnel with trade $k(k\in Kp)$ required for performing operation O_{ij}. The transfer time of personnel with trade $k(k\in Kp)$ between operation O_{eg} and operation O_{ij} is denoted as ΔP^k_{ijeg}, $\Delta P^k_{ijeg}=D(W_{ij},W_{eg})/vp_k$, where $D(W_{ij},W_{eg})$ denotes the distance between work station space W_{ij} and W_{eg}, vp_k denotes the transfer velocity of personnel with trade $k(k\in Kp)$.

(iii) Support equipment constraints

There are two kinds of support equipment, i.e., sharing equipment Ke_s and exclusive equipment Ke_u. Ke denotes the set of support equipment trade types, $Ke=Ke_s\cup Ke_u$, Le_k denotes the set of support equipment of the type $k(k\in Ke)$, $Le_k=\{1,2,\cdots,|Le_k|\}$. $\lambda^p_{kl}=1$ denotes the lth support equipment of the type $k(k\in Ke)$ which can reach the pth parking spot, otherwise, $\lambda^p_{kl}=0$. Equipment can only support the aircraft which are within the range of their pipelines. re_{ijk} denotes the number of support equipment of the type $k(k\in Ke)$ required for performing operation O_{ij}, $re_{ijk}\in\{0,1\}$. The transfer time of the support equipment of the type k between operation O_{eg} and operation O_{ij} is denoted as ΔE^k_{ijeg}, which consists of setup time of the support equipment of the type k (denoted as u_k), and moving time of the support equipment from work station space W_{ij} to W_{eg} (denoted as Tr^k_{ijeg}, $Tr^k_{ijeg}=D(W_{ij},W_{eg})/ve_k$, where ve_k denotes the transfer velocity of equipment). The setup time is considered both before and after performing operation, so $\Delta E^k_{ijeg}=Tr^k_{ijeg}+2u_k$. Fig. 2 shows a sketch map of personnel and equipment transfer process on flight deck of the Admiral Kuznetsov aircraft carrier.

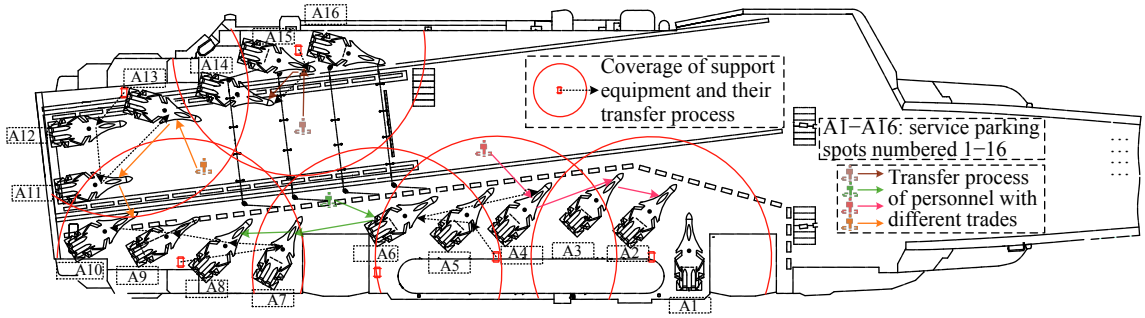


Fig. 2 Resources on flight deck of the Admiral Kuznetsov aircraft carrier

(iv) Work station space constraints

These constraints are considered only for the work station with limited space (e.g., cockpit) for personnel. Ks denotes the set of work station space types, and ns_{ik} denotes the maximum number of personnel who are performing concurrently in the kth $(k\in Ks)$ type work sta-

tion space of the ith aircraft. Set $rs_{ijk}=1$ if operation O_{ij} is performed in work station space of the type $k(k\in Ks)$, otherwise, $rs_{ijk}=0$.

(v) Supply resource constraints

Kw denotes the set of supply resource types. Lw_k denotes the maximum number of aircraft that the supply resource of type $k(k\in Kw)$ can support at the same time, and set $rw_{ijk}=1$ if the supply resource of type $k(k\in Kw)$ is required while performing O_{ij}, otherwise, $rw_{ijk}=0$.

4. Mathematical formulation of MAISPTT

4.1 Problem assumptions

(i) Each operation cannot be interrupted during its execution.

(ii) The duration of each operation is deterministic.

(iii) All kinds of resources are available for each service parking spot.

4.2 Decision variables

Besides $S_{ij},E_{ij},\forall(i,j)\in J$, the other relevant decision variables used in the model are as follows:

$$Xp_{ijkl}=\begin{cases}1, \text{ operation } O_{ij} \text{ is allocated to}\\ \quad \text{personnel } l \text{ with trade } k\\ 0, \text{ otherwise}\end{cases}$$

$$Xe_{ijkl}=\begin{cases}1, \text{ operation } O_{ij} \text{ is allocated to}\\ \quad \text{support equipment } l \text{ of the type } k\\ 0, \text{ otherwise}\end{cases}$$

$$Zp_{ijeg}^{kl}=\begin{cases}1, \text{ personnel } l \text{ with trade } k \text{ transfers}\\ \quad \text{to perform operation } O_{eg} \text{ after}\\ \quad \text{completing operation } O_{ij}\\ 0, \text{ otherwise}\end{cases}$$

$$Ze_{ijeg}^{kl}=\begin{cases}1, \text{ support equipment } l \text{ of the type } k\\ \quad \text{transfers to perform operation } O_{eg}\\ \quad \text{after completing operation } O_{ij}\\ 0, \text{ otherwise}\end{cases}$$

4.3 Mathematical expression of MAISPTT

MAISPTT is formulated as a mixed integer mathematical programming model:

$$\min\ C_{\max} \tag{1}$$

$$S_{i1}\geqslant Ex_i,\ \ \forall i\in I \tag{2}$$

$$S_{ij}\geqslant S_{ih}+d_{ih},\ \ \forall(i,h)\in \Psi_{ij};\forall(i,j)\in J \tag{3}$$

$$\sum_{i\in I}\sum_{(i,j)\in A_{it}} rw_{ijk}\leqslant Lw_k,\ \ \forall k\in Kw;\forall t>0 \tag{4}$$

$$\sum_{(i,j)\in A_{it}} rs_{ijk}\sum_{k'\in Kp} rp_{ijk'}\leqslant ns_{ik},\ \ \forall i\in I;\forall k\in Ks;\forall t>0 \tag{5}$$

$$\sum_{(i,j)\in J}\sum_{k\in Ke}\sum_{l\in Le_k} Xe_{ijkl}\cdot(1-\lambda_{kl}^{p_i})=0,\ \ \forall i\in I \tag{6}$$

$$E_{ij}+\Delta P_{ijeg}^{k}\leqslant S_{eg}+BM\cdot(1-Zp_{ijeg}^{kl}),\\ \forall k\in Kp;\forall l\in Lp_k;\forall(i,j),(e,g)\in J \tag{7}$$

$$E_{ij}+\Delta E_{ijeg}^{k}\leqslant S_{eg}+BM\cdot(1-Ze_{ijeg}^{kl}),\\ \forall k\in Ke_u\wedge(\forall k\in Ke_s,i\neq e);\\ \forall l\in Le_k;\forall(i,j),(e,g)\in J \tag{8}$$

$$\sum_{(i,j)\in A_{it}} re_{ijk}\cdot\operatorname{sgn}\left(\sum_{(i,j)\in A_{it}} Xe_{ijkl}\right)=\\ \sum_{(i,j)\in A_{it}} Xe_{ijkl}\cdot\operatorname{sgn}\left(\sum_{(i,j)\in A_{it}} re_{ijk}\right),\ \ \forall i\in I;\forall k\in Ke_s;\forall l\in Le_k \tag{9}$$

$$\sum_{l\in Lp_k} Xp_{ijkl}=rp_{ijk},\ \ \forall(i,j)\in J;\forall k\in Kp \tag{10}$$

$$\sum_{l\in Le_k} Xe_{ijkl}=re_{ijk},\ \ \forall(i,j)\in J;\forall k\in Ke \tag{11}$$

$$Zp_{ijeg}^{kl}\leqslant Xp_{ijkl}\cdot Xp_{egkl},\\ \forall(i,j),(e,g)\in J;\forall k\in Kp;\forall l\in Lp_k \tag{12}$$

$$Ze_{ijeg}^{kl}\leqslant Xe_{ijkl}\cdot Xe_{egkl},\\ \forall(i,j),(e,g)\in J;\forall k\in Ke;\forall l\in Le_k \tag{13}$$

$$Xp_{ijkl},Xe_{ijk'l'},Zp_{ijeg}^{kl},Ze_{ijeg}^{k'l'}=\{0,1\}\\ \forall k\in Kp;\forall l\in Lp_k;\forall k'\in Ke;\\ \forall l'\in Le_{k'};\forall(i,j),(e,g)\in J. \tag{14}$$

Formula (1) represents that the objective is to minimize the makespan $C_{\max}$ of the aircraft fleet. Constraint (2) indicates that aircraft i cannot start its operations until it is chocked and chained in the service parking spot at the released time Ex_i. Constraint (3) denotes the precedence constraints of operations. O_{ij} cannot be performed earlier than its immediate predecessors. Constraint (4) states that the total number of aircraft on the flight deck that are supported by supply resource of the type $k(k\in Kw)$ concurrently at the time period t is smaller than the maximum number Lw_k. Constraint (5) indicates that the number of personnel who are performing concurrently in the kth $(k\in Ks)$ type work station space of the ith aircraft at the time period t is smaller than the maximum number ns_{ik}. Constraint (6) denotes that equipment can only support the aircraft which are within the range of their pipelines. Constraint (7) represents that if operation O_{ij} precedes op-

eration O_{eg} and they are allocated to the same personnel l with trade $k(k \in Kp)$, the operation O_{eg} cannot be performed until this personnel transfers to work station space W_{eg} after completing operation O_{ij}, where BM is a large enough positive number. Constraint (8) is related to the transfer time sequence relationship between two adjacent operations, which are allocated to the same exclusive support equipment, or the same sharing support equipment while these two operations do not belong to the same aircraft. The transfer time sequence is similar to that in constraint (7). Constraint (9) indicates that if more than one operation of aircraft i require a sharing equipment of the type $k(k \in Ke_s)$ at any time period, these operations are allocated to the same one. Constraints (10) and (11) together make sure that the number of personnel or equipment required for performing operation O_{ij} is equal to the total number of personnel or equipment allocated to it respectively. Constraints (12) and (13) express relationships between decision variables. Constraint (14) indicates the values of decision variables.

5. The proposed DPMOGA

MAISPTT is a large-scale combinatorial optimization problem where precedence constraints and four renewable resources constraints need to be premeditated. In this paper, MAISPTT is considered as a resource-constrained multi-project scheduling problem with transfer times, and it is an NP-hard optimization problem. Hartmann et al. [21] pointed out that the most successful approaches for RCPSPs were meta-heuristics, one of which was GA. In this section, we describe DPMOGA to solve MAISPTT.

The flowchart of the DPMOGA is shown in Fig. 3.

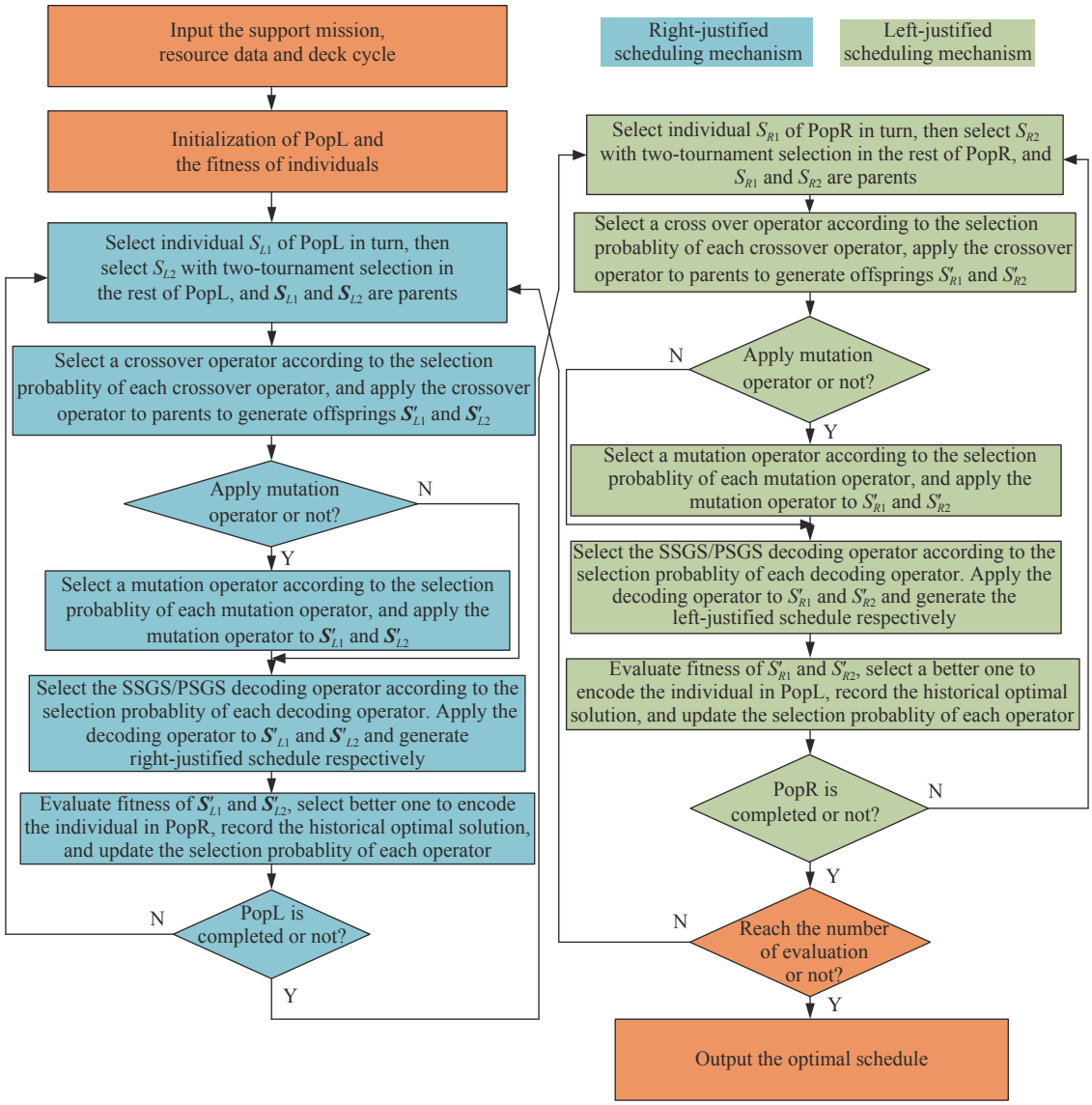


Fig. 3 Flowchart of DPMOGA

A left population PopL and a right population PopR are involved in our algorithm. In PopL, a right-justified scheduling mechanism is adopted to generate the right-justified schedule and the starting time of each operation in the schedule is used to encode the individual in PopL, while in PopR, a left-justified scheduling mechanism is adopted to generate the left-justified schedule and the ending time of each operation in the schedule is used to encode the individual in PopL.

Various types of genetic operators are proposed in our algorithm. In addition, an adaptive operator selection mechanism is set to choose the best performing genetic operator according to the performance of operators at each iteration. The fitness of individual S is denoted as f_S, which can be represented by the $C_{\max}$ of the generated schedule, and a lower $C_{\max}$ indicates a better fitness.

Compared with the GA, the DPMOGA is mainly improved in the following aspects.

(i) The dual population structure [18] is adopted, and the evolution process of population is divided into the left-justified scheduling mechanism and the right-justified scheduling mechanism. The double justification (DJ) technology is applied to all individuals in the population to enhance the algorithm's performance.

(ii) The encodings of individuals are modified by the starting/ending time of operations, which promises that one schedule can be represented by one encoding. Additionally, encodings are adjusted between the starting and the ending time of the operations of the scheduling plan, which increases the diversity of population.

(iii) Different genetic operators behave differently during the execution of the algorithm, and the best performing genetic operator will be selected by the adaptive operators selection mechanism for the evolution of the population. This method increases the stability and the probability of the DPMOGA.

5.1 Solutions encoding

Kolisch and Hartmann [22] distinguished five different schedule encodings, but the activity-list (AL) encoding and the random-key (RK) encoding are the most widespread. Debels et al. [23] indicated that the RK leads to promising results due to the use of the topological ordering (TO) notation. In our algorithm, the RK encoding modified by the starting/ending time of operations is adopted. Compared with AL and RK, our encoding promises that one schedule can be represented by one encoding. Set the encoding of individual $\boldsymbol{S}$ as $\boldsymbol{S} = [RK_{11}, RK_{12}, \cdots, RK_{n|J_n|}]$, where RK_{ij} indicates the priority of operation O_{ij}.

(i) For PopL, after generating the right-justified schedule, we encode the individual $\boldsymbol{S}_R$ in PopR using the starting time of each operation in the schedule, i.e., $\boldsymbol{S}_R = [S_{11}, S_{12}, \cdots, S_{n|J_n|}]$.

(ii) For PopR, after generating the left-justified schedule, we encode the individual $\boldsymbol{S}_L$ in PopL using the ending time of each operation in the schedule, i.e., $\boldsymbol{S}_L = [E_{11}, E_{12}, \cdots, E_{n|J_n|}]$.

5.2 Crossover operators

We adopt four kinds of crossover operators, which are described with taking PopL as an example.

Crossover 1: One-point crossover operator. An integer pos is generated randomly, where $1 < pos < |\boldsymbol{S}_L|$, then encodings of parents after the posth encoding are exchanged to generate offsprings.

Crossover 2: Two-point crossover operator. Integer $pos1$ and $pos2$ are generated randomly, where $1 < pos1 < pos2 < |\boldsymbol{S}_L|$, then encodings of parents between the $pos1$th encoding and the $pos2$th encoding are exchanged to generate offsprings.

Crossover 3: Arithmetic crossover operator. When this crossover operator is applied to parents $\boldsymbol{S}_{L1}$ and $\boldsymbol{S}_{L2}$, the offsprings $\boldsymbol{S}'_{L1}$ and $\boldsymbol{S}'_{L2}$ are calculated as

$$\begin{cases} \boldsymbol{S}'_{L1} = a\boldsymbol{S}_{L1} + b\boldsymbol{S}_{L2} \\ \boldsymbol{S}'_{L2} = b\boldsymbol{S}_{L1} + a\boldsymbol{S}_{L2} \end{cases} \tag{15}$$

where a and b are random numbers, $a, b \in [-0.5, 1.5], a+b=1$. The values of a and b are not limited to [0,1], which ensures that the search area of the arithmetic crossover operator covers the parents' neighborhood more widely.

Crossover 4: Direction-based heuristic crossover operator. The origin version of this operator was proposed by Wang et al. [24], who indicated that there is a great chance to produce better offsprings to improve the convergence speed of the algorithm. Denote $\boldsymbol{C} = \boldsymbol{S}_{L2} - \boldsymbol{S}_{L1}$, r_{1i} and $r_{2i} (i = 1, 2, \cdots, J)$ are random numbers limited to [0,1], $\boldsymbol{C}_1 = [r_{1i}c_i]_{1\times J}$, $\boldsymbol{C}_2 = [r_{2i}c_i]_{1\times J}$, step $\lambda = 1$, the offsprings $\boldsymbol{S}'_{L1}$ and $\boldsymbol{S}'_{L2}$ are calculated as

$$\begin{cases} \boldsymbol{S}'_{L1} = \boldsymbol{S}_{L1} + \lambda\boldsymbol{C}_1 \\ \boldsymbol{S}'_{L2} = \boldsymbol{S}_{L2} + \lambda\boldsymbol{C}_2 \end{cases}. \tag{16}$$

5.3 Adaptive mutation probability and mutation operators

5.3.1 Adaptive mutation probability

Our adaptive mutation probability P_m is based on mean fitness variance:

$$\sigma^2 = \frac{1}{NP}\sum_{j=1}^{NP}\left(\frac{f_j - f_{\text{avg}}}{D}\right)^2 \tag{17}$$

where NP is the population size, and D represents the maximum deviation between the fitness of individuals in the population and the average fitness, $D = \max\limits_{1\leqslant j\leqslant NP}\{|f_j - f_{\text{avg}}|\}$. P_m is calculated as

$$P_m = \begin{cases} k, & \sigma^2 < \sigma_d^2 \\ P_{m0}, & \text{otherwise} \end{cases} \tag{18}$$

where k is a random number limited to [0.3,0.6], and σ_d^2 is a threshold value. P_{m0} is a relatively small initializing mutation probability. When the individual difference in the population decreases, the mean fitness variance of the population decreases accordingly. At this time, increasing the individual mutation probability has an advantage of helping the algorithm jumping over the local optimal solution. In addition, a random number *rand* is generated in each cycle, $rand \in [0,1]$, if $rand < P_m$, a mutation operator is selected and applied to individuals.

5.3.2 Mutation operators

We adopt four kinds of mutation operators in our algorithm.

Mutation 1: One-point mutation operator. An encoding is selected randomly and then modified to $(rand - 0.5) \cdot RK_{\max}$, where $RK_{\max}$ is the maximum of encodings.

Mutation 2: One-operation mutation operator. An operation O_{ij} with the latest ending time t_1 of its predecessors and the earliest starting time t_2 of its successors in the schedule, is selected randomly. The encoding of operation O_{ij} is modified to a random number limited to $[t_1 + d_{ij}, t_2]$ (for individual in PopL) or a random number limited to $[t_1, t_2 - d_{ij}]$ (for individual in PopR).

Mutation 3: Multi-operation of one aircraft mutation operator. $Na(Na \in Z[|J_i|/3, |J_i| \times 2/3])$ operations of a single aircraft i are selected randomly, then each encoding of them is modified by the method in Mutation 2.

Mutation 4: Project reorganization mutation operator. Aircraft $i(i \in I)$ is selected randomly and the encodings of all the operations of this aircraft are initialized according to precedence constraints.

5.4 Decoding operators

Decoding is a key stage to conduct the mapping of individual encoding to schedule and evaluate the fitness by using SGS. The most widespread SGS in RCPSPs is SSGS or PSGS [25]. In our algorithm, the SSGS-based decoding operator and the PSGS-based decoding operator are proposed simultaneously. For PopL, the right-justified SSGS-based or right-justified PSGS-based decoding operator is applied to generate the right-justified schedule, while for PopR, the left-justified SSGS-based or the left-justified PSGS-based decoding operator is applied to generate the left-justified schedule. The related notations are described in Table 1.

Table 1 Related notations in decoding operators

Notation	Definition
C_g	The set of scheduled operations at stage g
D_g	The set of eligible operations at stage g
t_g	The earliest feasible scheduling time of stage g
A_g	The set of operations active in period t_g of stage g
t'_g	The earliest precedence-resource-feasible scheduling time of stage g
πLp_{kl}^t	0, if personnel l with trade $k(k \in Kp)$ is not performing or transferring in period t; 1, otherwise
πLe_{kl}^t	i, if support equipment l of the type $k(k \in Ke)$ is supporting aircraft i $(i = 1,2,\cdots,n)$ in period t; -1 if this support equipment is transferring; 0, otherwise
πLs_{ik}^t	1, if work station space of the type $k(k \in Ks)$ in aircraft i in period t is occupied; 0, otherwise
πLw_k^t	The remaining number of aircraft that supply resource of the type $k(k \in Kw)$ can support in period t
ES_{ij}	The earliest precedence-feasible starting time of operation O_{ij}
SP_{ij}	The earliest personnel-feasible starting time of operation O_{ij}
SP_{ijk}	The earliest personnel-feasible starting time of operation O_{ij} for personnel with trade $k(k \in Kp)$
SP_{ijkl}	The earliest personnel-feasible starting time of operation O_{ij} for personnel l with trade $k(k \in Kp)$
SE_{ij}	The earliest equipment-feasible starting time of operation O_{ij}
SE_{ijk}	The earliest equipment-feasible starting time of operation O_{ij} for support equipment of the type $k(k \in Ke)$
SE_{ijkl}	The earliest equipment-feasible starting time of operation O_{ij} for support equipment l of the type $k(k \in Ke)$
SS_{ij}	The earliest space-feasible starting time of operation O_{ij}
SW_{ij}	The earliest supply resource-feasible starting time of operation O_{ij}
ERS_{ij}	The earliest precedence-resource-feasible starting time of operation O_{ij}
lp_k^t	The set of personnel with trade $k(k \in Kp)$ that can perform operation to be scheduled at period t
le_k^t	The set of support equipment of the type $k(k \in Ke)$ that can perform operation to be scheduled at period t
$Le_k^{p_i}$	The set of support equipment of the type $k(k \in Ke)$ that can reach the service parking spot p_i, $Le_k^{p_i} = \{l \mid \lambda_{kl}^{p_i} = 1, l \in Le_k\}$
RP_k^t	The set of personnel with trade $k(k \in Kp)$ that are not performing or transferring at period t
RE_k^t	The set of support equipment of the type $k(k \in Ke)$ that are not performing or transferring at period t
AP_{kl}	The set of scheduled operations of the personnel l with trade $k(k \in Kp)$
AE_{kl}	The set of scheduled operations of the support equipment l of the type $k(k \in Ke)$
TT_{kl}	The accumulated transfer distance of the personnel l with trade $k(k \in Kp)$
TR_{kl}	The total performing time remaining in the cover area of the support equipment l of the type $k(k \in Ke)$

5.4.1 Left-justified SSGS-based decoding scheme

There are $|J|$ stages totally in SSGS. C_g is initialized with the dummy starting operation of each aircraft, and the starting time of operations in C_g are set as the released time of corresponding aircraft. At each stage g, the set of eligible operations is calculated, from which the operation to be scheduled $O_{i^*j^*}$ with the highest priority (minimal RK) is chosen.

Then the earliest precedence-resource-feasible starting time of operation $O_{i^*j^*}$ is initialized by the earliest precedence-feasible starting time of it. To arrange the operation which satisfies the constraints of personnel, support equipment, work station space and supply resource in the duration, the time $ERS_{i^*j^*}$ is postponed until the earliest resource-feasible starting time of the four kinds of resource is equal, i.e., $SP_{i^*j^*}=SE_{i^*j^*}=SS_{i^*j^*}=SW_{i^*j^*}$. Afterwards, the starting time $S_{i^*j^*}$ is determined as the earliest precedence-resource-feasible time. In order to allocate personnel and support equipment for the operation after determining $S_{i^*j^*}$, the personnel allocation rule and the support equipment allocation rule simultaneously are proposed as follows.

Rule 1: Personnel allocation rule based on the minimum accumulated transfer distance first (MATDF). If the operation to be scheduled requires personnel with trade $k(k\in Kp)$, the accumulated transfer distance TT_{kl} of each personnel l with trade k that can support the operation is calculated, then personnel are allocated for the operation with the ascending order of TT_{kl}.

Rule 2: Support equipment allocation rule based on the minimum total performing time remaining in the cover area (MTRCA). If the operation to be scheduled requires support equipment of the type $k(k\in Ke)$, the total performing time remaining in the cover area of each support equipment l of the type k that can support the operation TR_{kl} is calculated as follows:

$$TR_{kl}=\sum_{(i,j)\in J'_{kl}} d_{ij} \tag{19}$$

where the set of remaining operations is $J'_{kl}=\{(i,j)\,|\,(i,j)\in J-C_g,\lambda_{kl}^{p_i}=1,re_{ijk}\geqslant 0\}$.

At the end of this loop, all the state parameters and the set of scheduled operations are updated for the next stage. The pseudo code of the left-justified SSGS-based decoding scheme is shown in Algorithm 1, Algorithm 2 and Algorithm 3.

Algorithm 1 Algorithm of the left-justified SSGS-based decoding scheme

Input Priority of operations: RK_{ij}

Output Temporal schedule $\{S_{ij},E_{ij}\}$, scheduling plan $\left\{Xp_{ijkl},Xe_{ijkl},Zp_{ijeg}^{kl},Ze_{ijeg}^{kl}\right\}$

01:**Initialize** $\pi Lp_{kl}^t,\pi Le_{kl}^t,\pi Lw_k^t,\pi Ls_{ik}^t,TT_{kl},TR_{kl},S_{i1}:=Ex_i$, $E_{i1}:=Ex_i,g:=1,C_g:=\cup_{i\in I}\{(i,1)\}$

02: **While** $|C_g|<|J|$

03: $D_g:=\left\{(i,j)\middle|(i,j)\notin C_g,P_{ij}\subseteq C_g\right\},(i^*,j^*):=\min_{O_{ij}\in D_g}\left\{(i,j)\middle|RK_{ij}=\inf_{O_{pq}\in D_g}(RK_{pq})\right\}$, $ES_{i^*j^*}:=\max\left\{E_{ij}\middle|(i,j)\in P_{i^*j^*}\right\}$

04: $ERS_{i^*j^*}:=ES_{i^*j^*}$

05: **Repeat**

06: Calculate $SP_{i^*j^*}$, lp_k^t, $\forall k\in Kp\wedge rp_{i^*j^*k}>0(t=SP_{i^*j^*})$ using Algorithm 2

07: Calculate $SE_{i^*j^*}$, le_k^t, $\forall k\in Ke\wedge re_{i^*j^*k}>0(t=SE_{i^*j^*})$ using Algorithm 3

08: $SS_{i^*j^*}:=\min\{t|t\geqslant ERS_{i^*j^*},rs_{i^*j^*k}\cdot\pi Ls_{i^*k}^{\tau}\neq 1,k\in Ks,\tau\in[t,t+d_{i^*j^*})\}$

09: $SW_{i^*j^*}:=\min\{t|t\geqslant ERS_{i^*j^*},rw_{i^*j^*k}\pi Lw_k^{\tau},k\in Kw,\ \tau\in[t,t+d_{i^*j^*})\}$

10: $ERS_{i^*j^*}:=\max\{SP_{i^*j^*},SE_{i^*j^*},SS_{i^*j^*},SW_{i^*j^*}\}$

11: **Until** $SP_{i^*j^*}=SE_{i^*j^*}=SS_{i^*j^*}=SW_{i^*j^*}$

12: $S_{i^*j^*}:=ERS_{i^*j^*}$, $E_{i^*j^*}:=ERS_{i^*j^*}+d_{i^*j^*}$

13: **For** $\forall k\in Kp\wedge rp_{i^*j^*k}>0$

14: Arrange all personnel $l\in lp_k^t(t=SP_{i^*j^*})$ in the ascending order of TT_{kl}

15: **For** $p=1:rp_{i^*j^*k}$ *Db* Let l^* be the personnel in position p in the set $lp_k^t(t=SP_{i^*j^*})$, $Xp_{i^*j^*kl^*}:=1$

End For

16: **End For**

17: **For** $\forall k\in Ke\wedge re_{i^*j^*k}>0$ select $l^*\in le_k^t(t=SE_{i^*j^*})$, according to MTPTRCA, $Xe_{i^*j^*kl^*}:=1$

End For

18: Update $Zp_{ijeg}^{kl},Ze_{ijeg}^{kl},\pi Lp_{kl}^t,\pi Le_{kl}^t,\pi Lw_k^t,\pi Ls_{ik}^t,TT_{kl},TR_{kl}$

19: $g:=g+1$, $C_g:=C_{g-1}\cup\{(i^*,j^*)\}$

20: **End While**

Algorithm 2 Algorithm of calculating $SP_{i^*j^*}$, lp_k^t, $\forall k\in Kp\wedge rp_{i^*j^*k}>0(t=SP_{i^*j^*})$

Input (i^*,j^*) $ERS_{i^*j^*}$

Output $SP_{i^*j^*}$, lp_k^t, $\forall k\in Kp\wedge rp_{i^*j^*k}>0(t=SP_{i^*j^*})$

01:**Initialize** $SP_{i^*j^*k}:=k,\forall k\in Kp$ $k:=1$, $SP_{i^*j^*}:=ERS_{i^*j^*}$

02:**Repeat**

03: **If** $k>|Kp|$ **Then** $k:=1$

End If

04: **If** $rp_{i^*j^*k}=0$ **Then** $SP_{i^*j^*k}:=ERS_{i^*j^*},k:=k+1$

05: **Else**

06: $RP_k^t=\{l|\pi Lp_{kl}^t=0,l\in Lp_k,t=ERS_{i^*j^*}\}$, **Initialize** $lp_k^t=\varnothing(\forall t>0)$

07: **For** $\forall l\in RP_k^t(t=ERS_{i^*j^*})\wedge RP_k^t\neq\varnothing$

08: Judge whether personnel l can perform operation (i^*,j^*) according to Fig. 4 and calculate $lp_k^t(t=ERS_{i^*j^*})$

09: **End For**

10: **If** $|lp_k^t|\geqslant rp_{i^*j^*k}(t=ERS_{i^*j^*})$ **Then** $SP_{i^*j^*k}:=ERS_{i^*j^*}$, $k:=k+1$

11: **Else** $ERS_{i^*j^*}:=ERS_{i^*j^*}+1$

12: **End If**
13: **End If**
14:**Until**$(k=|Kp|)\wedge(SP_{i^*j^*k'}=SP_{i^*j^*k''},\forall k',k''\in Kp)$
15:$SP_{i^*j^*}:=ERS_{i^*j^*}$

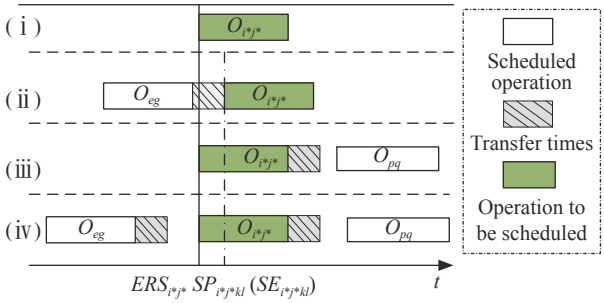


Fig. 4 Scheduling timeline of personnel or support equipment

Algorithm 3 Algorithm of calculating $SE_{i^*j^*}$, le_k^t, $\forall k\in Ke\wedge re_{i^*j^*k}>0(t=SE_{i^*j^*})$

Input (i^*,j^*) $ERS_{i^*j^*}$

Output $SE_{i^*j^*}$, le_k^t, $\forall k\in Ke\wedge re_{i^*j^*k}>0(t=SE_{i^*j^*})$

01:**Initialize** $SE_{i^*j^*k}:=k,\forall k\in Ke,k:=1,SE_{i^*j^*}:=ERS_{i^*j^*}$
02: **Repeat**
03: **If** $k>|Ke|$ **Then** $k:=1$
End If
04: **If** $re_{i^*j^*k}=0$ **Then** $SE_{i^*j^*k}:=ERS_{i^*j^*},k:=k+1$
05: **Else**
06: **Initialize** $le_k^t=\emptyset(\forall t>0)$
07: **If** $(k\in Ke_s)\wedge(\{l|\pi Le_{kl}^t=i^*,l\in Le_k^{p_{i^*}},t\in[ERS_{i^*j^*},ERS_{i^*j^*}+d_{i^*j^*})\}\neq\emptyset)$ **Then**
08: $SE_{i^*j^*k}:=ERS_{i^*j^*},le_k^{t'}:=\{l|\pi Le_{kl}^t=i^*,l\in Le_k^{p_{i^*}},t\in[ERS_{i^*j^*},ERS_{i^*j^*}+d_{i^*j^*})\}(t'=ERS_{i^*j^*})$
09: $k:=k+1$
10: **Else**
11: $RE_k^t=\{l|\pi Le_{kl}^t=0,l\in Le_k^{p_{i^*}},t=ERS_{i^*j^*}\}$
12: **For** $\forall l\in RE_k^t(t=ERS_{i^*j^*})\wedge RE_k^t\neq\emptyset$
13: Judge whether support equipment l can support operation (i^*,j^*) according to Fig. 4 and calculate $le_k^t(t=ERS_{i^*j^*})$
14: **End For**
15: **If** $|le_k^t|\geqslant re_{i^*j^*k}(t=ERS_{i^*j^*})$ **Then** $SE_{i^*j^*k}:=ERS_{i^*j^*}$, $k:=k+1$
16: **Else** $ERS_{i^*j^*}:=ERS_{i^*j^*}+1$
17: **End If**
18: **End If**
19: **End If**
20: **Until** $(k=|Ke|)\wedge(SE_{i^*j^*k'}=SE_{i^*j^*k''},\forall k',k''\in Ke)$
21: $SE_{i^*j^*}:=ERS_{i^*j^*}$

The difference for scheduling timeline between personnel and support equipment is transfer times in Fig. 4. The procedure to judge whether personnel l can perform operation O_{ij} and to calculate $lp_k^t(t=ERS_{i^*j^*})$ is described as follows, and the procedure to judge support equipment and to calculate $le_k^t(t=ERS_{i^*j^*})$ is similar.

Situation 1: This personnel does not support any scheduled operations. It is feasible to allocate this personnel to perform operation $O_{i^*j^*}$, and $SP_{i^*j^*kl}:=ERS_{i^*j^*}$ $lp_k^t:=lp_k^t\bigcup\{l\}(t=SP_{i^*j^*kl})$.

Situation 2: The last scheduled operation of this personnel is O_{eg} and $ERS_{i^*j^*}>E_{eg}$. It is feasible to allocate this personnel to perform operation $O_{i^*j^*}$, and $SP_{i^*j^*kl}:=\max\{E_{eg}+\Delta P_{egi^*j^*}^k,ERS_{i^*j^*}\}$, $lp_k^t:=lp_k^t\bigcup\{l\}(t=SP_{i^*j^*kl})$.

Situation 3: The first scheduled operation of this personnel is O_{pq} and $ERS_{i^*j^*}+d_{i^*j^*}+\Delta P_{i^*j^*pq}^k\leqslant S_{pq}$. It is feasible to allocate this personnel to perform operation $O_{i^*j^*}$, and $SP_{i^*j^*kl}:=ERS_{i^*j^*}$, $lp_k^t:=lp_k^t\bigcup\{l\}(t=SP_{i^*j^*kl})$.

Situation 4: $ERS_{i^*j^*}$ is between the starting time of scheduled operations O_{eg} and O_{pq}, and $E_{eg}+\Delta P_{egi^*j^*}^k+d_{i^*j^*}+\Delta P_{i^*j^*pq}^k\leqslant S_{pq}$. It is feasible to allocate this personnel to perform operation $O_{i^*j^*}$, and $SP_{i^*j^*kl}:=\max\{E_{eg}+\Delta P_{egi^*j^*}^k,ERS_{i^*j^*}\}$, $lp_k^t:=lp_k^t\bigcup\{l\}(t=SP_{i^*j^*kl})$.

5.4.2 Left-justified PSGS-based decoding scheme

The pseudo code of the left-justified PSGS-based decoding scheme is shown in Algorithm 4. In this operator, there are $|J|$ stages at most. At the beginning, when g=1, the sets A_g and C_g are initialized with the dummy starting operation of each aircraft and null set respectively, and the starting time of operations in A_g are set as the released time of corresponding aircraft. Then at each stage g, if all the processors of an operation are scheduled, the operation can be added into set D_g. t_g is calculated as the earliest ending time of operations in set A_{g-1}. All operations in D_g are arranged in the ascending order of RK_{ij}. For each operation $O_{i^*j^*}\in D_g$, if required resources are feasible at period t_g, its starting time $S_{i^*j^*}$ is set as the earliest time when all required personnel and support equipment are transferred to work station $W_{i^*j^*}$. Since $S_{i^*j^*}\geqslant t_g$, it is necessary to determine whether any other personnel or support equipment can perform operation $O_{i^*j^*}$ at period $S_{i^*j^*}$. The personnel allocation rule and support equipment allocation rule are also Rule 1 and Rule 2 simultaneously.

Algorithm 4 Algorithm of the left-justified PSGS-based decoding scheme

Input Priority of operations: RK_{ij}

Output Temporal schedule $\{S_{ij},E_{ij}\}$, scheduling plan $\{Xp_{ijkl},Xe_{ijkl},Zp_{ijeg}^{kl},Ze_{ijeg}^{kl}\}$

01:**Initialize** $\pi Lp_{kl}^t,\pi Le_{kl}^t,\pi Lw_k^t,\pi Ls_{ik}^t,TT_{kl},TR_{kl},AP_{kl},AE_{kl},A_1:=\cup_{i\in I}\{(i,1)\},C_1:=\emptyset,S_{i1}:=Ex_i,E_{i1}:=Ex_i,g:=$
02: **While** $|C_{g-1}\cup A_{g-1}|<|J|$
03: $t_g:=\min\{S_{ij}+d_{ij}|(i,j)\in A_{g-1}\},A_g:=A_{g-1}-\{(i,j)|(i,j)\in A_{g-1}\wedge t_g=E_{ij}\}$
04: $C_g:=C_{g-1}\cup\{(i,j)|(i,j)\in A_{g-1}\wedge t_g=E_{ij}\}$, $D_g:=$

$\{(i,j)|\ (i,j)\notin A_g, P_{ij}\subseteq C_g\}$

05: Arrange all operations $(i,j)\in D_g$ in the ascending order of RK_{ij}, $q:=1$

06: **While** $q<|D_g|$

07: Let (i^*,j^*) be the operation in position q in D_g, **Initialize** $lp_k^t:=\varnothing$, $SP_{i^*j^*k}:=t_g,\forall k\in Kp, le_{k'}^t:=\varnothing$, $SE_{i^*j^*k'}:=t_g,\forall k'\in Ke$

08: **For** $\forall k\in Ks$

If $rs_{i^*j^*k}\cdot\pi Ls_{i^*k}^{t_g}=1$ **Then** $q:=q+1$, **Turn to** line 06

End If

End For

09: **For** $\forall k\in Kw$

If $rw_{i^*j^*k}>\pi Lw_k^{t_g}$ **Then** $q:=q+1$, **Turn to** line 06

End If

End For

10: **For** $\forall k\in Kp\wedge rp_{i^*j^*k}>0$

11: $RP_k^{t_g}:=\{l|\pi Lp_{kl}^{t_g}=0, l\in Lp_k\}$

12: **If** $|RP_k^{t_g}|\geqslant rp_{i^*j^*k}$ **Then**

13: **For** $\forall l\in RP_k^{t_g}$ $SP_{i^*j^*kl}:=\max\left\{\max_{(e,g)\in AP_{kl}}\{E_{eg}+\Delta P_{egi^*j^*}^k\}, t_g\right\}$

End For

14: Arrange all personnel $l\in RP_k^{t_g}$ in the ascending order of $SP_{i^*j^*kl}$, delete the last $|RP_k^{t_g}|-rp_{i^*j^*k}$ personnel

15: $SP_{i^*j^*k}:=\max_{l\in RP_k^{t_g}}\{SP_{i^*j^*kl}\}, lp_k^t:=RP_k^{t_g}(t=SP_{i^*j^*k})$

16: **Else** $q:=q+1$, **Turn to** line 06

17: **End If**

18: **End For**

19: **For** $\forall k\in Ke\wedge re_{i^*j^*k}>0$

20: **If** $(k\in Ke_s)\wedge\left(\left\{l|\pi Le_{kl}^{t_g}=i^*, l\in Le_k^{p_{i^*}}\right\}\neq\varnothing\right)$ **Then** $SE_{i^*j^*k}:=t_g, le_k^t:=\left\{l|\pi Le_{kl}^{t_g}=i^*, l\in Le_k^{p_{i^*}}\right\}(t=SE_{i^*j^*k})$

21: **Else**

22: $RE_k^{t_g}:=\{l|\pi Le_{kl}^t=0, l\in Le_k^{p_{i^*}}\}$

23: **If** $|RE_k^{t_g}|\geqslant re_{i^*j^*k}$ **Then**

24: **For** $\forall l\in RE_k^{t_g}$ $SE_{i^*j^*kl}:=\max\left\{\max_{(e,g)\in AE_{kl}}\left\{E_{eg}+\Delta E_{egi^*j^*}^k\right\}, t_g\right\}$

End For

25: $SE_{i^*j^*k}:=\min_{l\in RE_k^{t_g}}\{SE_{i^*j^*kl}\}$, $l^*:=\{l|SE_{i^*j^*kl}=SE_{i^*j^*k}, l\in RE_k^{t_g}\}$, $le_k^t:=\{l^*\}(t=SE_{i^*j^*k})$

26: **Else** q:=q+1, **Turn to** line 06

27: **End If**

28: **End If**

29: **End For**

30: $S_{i^*j^*}:=\max_{k\in Kp,k'\in Ke}\{SP_{i^*j^*k}, SE_{i^*j^*k'}\}, E_{i^*j^*}:=S_{i^*j^*}+d_{i^*j^*}$, $t_g':=S_{i^*j^*}$ $A_g:=A_g\cup\{i^*,j^*\}$

31: **For** $\forall k\in Kp\wedge rp_{i^*j^*k}>0$ $lp_k^{t'_g}:=lp_k^t(t=SP_{i^*j^*k})$

End For

32: **For** $\forall k\in Ke\wedge re_{i^*j^*k}>0$ $le_k^{t'_g}:=le_k^t(t=SE_{i^*j^*k})$

End For

33: **For** $\forall k\in Kp\wedge rp_{i^*j^*k}>0$

34: **For** $\forall l\in RP_k^{t_g}-RP_k^{t'_g}$

If $\max_{(e,g)\in AP_{kl}}\left\{E_{eg}+\Delta P_{egi^*j^*}^k\right\}\leqslant t_g'$ **Then** $lp_k^{t'_g}:=lp_k^{t'_g}\cup\{l\}$

End If

End For

35: **End For**

36: **For** $\forall k\in Ke\wedge re_{i^*j^*k}>0$

37: **If** $(k\in Ke_s)\wedge\left(\left\{l|\pi Le_{kl}^{t_g}=i^*, l\in Le_k^{p_{i^*}}\right\}\neq\varnothing\right)$ **continue**

End If

38: **For** $\forall l\in RE_k^{t_g}-RE_k^{t'_g}$

If $\max_{(e,g)\in AE_{kl}}\left\{E_{eg}+\Delta E_{egi^*j^*}^k\right\}\leqslant t_g'$ **Then** $le_k^{t'_g}:=le_k^{t'_g}\cup\{l\}$

End If

End For

39: **End For**

40: **For** $\forall k\in Kp\wedge rp_{i^*j^*k}>0$

41: Arrange all personnel $l\in lp_k^{t'_g}$ in the ascending order of TT_{kl}

42: **For** $p=1:rp_{i^*j^*k}$

43: Let l^* be the personnel in position p in the set $lp_k^{t'_g}$, $Xp_{i^*j^*kl^*}:=1$

44: **End For**

45: **End For**

46: **For** $\forall k\in Ke\wedge re_{i^*j^*k}>0$ Select $l^*\in le_k^{t'_g}$ according to MTRCA, $Xe_{i^*j^*kl^*}:=1$

End For

47: Update Zp_{ijeg}^{kl}, Ze_{ijeg}^{kl}, πLp_{kl}^t, πLe_{kl}^t, πLw_k^t, πLs_{ik}^t, TT_{kl}, TR_{kl}, AP_{kl}, AE_{kl}, $q:=q+1$

48: **End While**

49: $g:=g+1$

50: **End While**

5.4.3 Right-justified SSGS/PSGS-based decoding scheme

Operations can also be scheduled oppositely by the right-justified scheduling mechanism, that is, all operations are shifted to the right except for the first and last operations, and the schedule starts from the final operation's immediate predecessors. In this way, the ending time is set as late as possible.

Contrary to the left-justified scheduling mechanism, the operation O_{ij} is added to C_g, if and only if all the successors of O_{ij} are scheduled. C_g is initialized with the

dummy ending operation of each aircraft, and the ending time of operations in C_g are set late enough to ensure that all operations start from a positive time. A larger RK indicates a higher priority in the right-justified scheduling mechanism. The personnel allocation rule and the support equipment allocation rule are also Rule 1 and Rule 2. Then after the right-justified scheduling plan $\{S_{ij}, E_{ij}\}$ is generated, the makespan can be reduced by subtracting a span ΔT^L from the whole schedule, where $\Delta T^L = \min_{i \in I_e} (S_{i1} - Ex_i)$, and the final right-justified scheduling plan is obtained as $(S_{ij}, E_{ij}) = (S_{ij}, E_{ij}) - \Delta T^L$.

5.5 Adaptive operators selection mechanism

This section describes how to select a genetic operator. Our selection mechanism is based on the mechanism in the consolidated optimization algorithm (COA) proposed by Elsayed et al. [12]. Nf denotes the type number of genetic operators, for crossover operators and mutation operators, Nf=4, for decoding operators, Nf=2. Let's take the crossover operators as an example. In the CSth cycle, the improvement rate of the ith crossover operator is denoted as $I_m(CS,i)(i=1,2,\cdots,Nf)$. Initialize $I_m(1,i) = 0(i=1,2,\cdots,Nf)$ at the beginning of the algorithm, once the ith crossover operator is applied to the individual S in the CSth cycle, $I_m(CS,i)$ is updated as

$$I_m(CS,i) = I_m(CS-1,i) + \frac{C_{\max,\text{old}} - C_{\max}}{C_{\max,\text{old}}} \tag{20}$$

where $C_{\max,\text{old}}$ is the minimal $C_{\max}$ in the last cycle. The improvement rate is updated in PopL or PopR respectively. Moreover, if $I_m(CS,i) \leqslant 0$, it is initialized to a small enough positive number $SM = 10^{-5}$. It is worth noting that there is a mutation probability before applying a mutation operator. If no mutation operator applies to the individual, the improvement rate will not update. The selection probability of the ith crossover operator in the next cycle is calculated as

$$\text{prob}(CS+1,i) = \max\left\{0.1, \min\left\{0.9, I_m(CS,i)\Big/\sum_{i=1}^{Nf} I_m(CS,i)\right\}\right\}. \tag{21}$$

Also, in the case that no improvement in the fitness after a crossover operator is applied to the individual, all the selection probabilities of each crossover operator in the next cycle are set to be equal.

5.6 Computational complexity analysis

The crossover operators, the mutation operators, and the decoding operators are key parts of the DPMOGA. The complexity of crossover and mutation operators is shown in Table 2. The computational complexity of DPMOGA is mainly reflected in decoding operators. According to [26], the complexity of SSGS and PSGS is both $O(|J|^2R)$, where R is the number of renewable resource types. In the MAISPTT, the decoding operation of PopL and PopR is consistent. The status of four resources have to be considered while finding feasible resources, and the complexities for finding personnel, support equipment, supply resources, and work station space are $O\left(|J|^2\sum_{k\in Kp}|Lp_k|\right)$, $O\left(|J|^2\sum_{k\in Ke}|Le_k|\right)$, $O\left(|J|^2\sum_{k\in Kw}|Lw_k|\right)$, and $O(|J|^2\times|Ks|)$ respectively.

Table 2 Complexity of genetic operators

Crossover operator		Mutation operator	
Crossover 1	$O(NP)$	Mutation 1	$O(NP)$
Crossover 2	$O((\|J\|-2)\times NP)$	Mutation 2	$O(NP)$
Crossover 3	$O(\|J\|\times NP)$	Mutation 3	$O(Na\times NP)$
Crossover 4	$O(\|J\|\times NP)$	Mutation 4	$O(\|J_i\|\times NP)$

6. Simulation experiments

6.1 Mission cases generation

The simulation cases are generated based on the Admiral Kuznetsov flight deck as shown in Fig. 2. The general AoN network is shown in Fig. 5. There are 19 real operations to be performed for each single aircraft except for the dummy starting operation (numbered 1) and the dummy ending operation (numbered 21). The numbers regarding the types of personnel, support equipment and supply resources are $|Kp|$=4, $|Ke|$=7, and $|Kw|$=5 respectively, and the required unit of personnel and support equipment for all the relevant operations is 1. Only one kind of work station space (cockpit) is considered, i.e., $|Ks|$=1, and the maximum number of personnel who are performing concurrently in cockpit is 1. In addition, the maximum number of aircraft that each supply resource can support at the same time is set as $[Lw_1, Lw_2, \cdots, Lw_5] = [9,14,2,4,6]$.

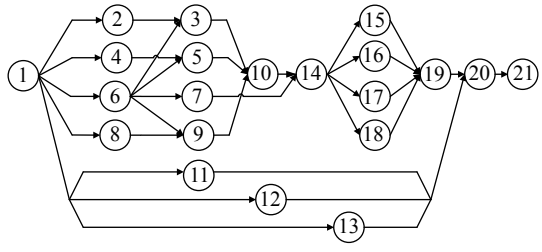


Fig. 5 AoN network of single-aircraft

According to [10], the configured number of personnel should be determined based on mission and experience. PS denotes the average personnel strength, and the number of personnel with trade $k(k \in Kp)$ is calculated as

$$|Lp_k| = \frac{PS \cdot \sum_{(i,j)\in J} rp_{ijk} d_{ij}}{C^{\mathrm{d}}_{\max}}. \tag{22}$$

In our simulation experiments, the deck cycle is set as $C^{\mathrm{d}}_{\max} = 80$ min.

The first five types support equipment (numbered 1−5) are fixed stations, and their configured numbers are $[|Le_1|, |Le_2|, \cdots, |Le_5|] = [7, 16, 6, 6, 8]$. Among these fixed stations, support equipment No.2 is sharing support equipment, while others are exclusive support equipment. Moreover, support equipment No.6 and No.7 are mobile equipment, which can be regarded as stationary equipment whose support coverage is the whole area of flight deck, and their configured number is sufficient and without constraints.

The personnel transfer velocity is set as $vp_k = 5$ km/h, $\forall k \in Kp$. For support equipment numbered 1−6, their transfer velocity is set as 3 km/h, and the setup time is set as $[u_1, u_2, \cdots, u_6] = [20, 10, 30, 30, 20, 20]$. The distance matrix between work stations, the requirements and duration of each operation are large-scale and not listed.

Three types of mission cases are described in Table 3. There are two average personnel strengths, PS=1.8 and PS=3.2 in each mission case.

Table 3 Mission cases

Mission case	Service parking spot number															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
	Aircraft type; released time/min															
Mission 1	A;0	B;0	C;0	C;0	D;0	D;0	D;0	D;0	—	—	—	—	—	—	—	—
Mission 2	A;0	B;0	A;0	C;0	C;0	C;0	C;0	D;0	D;8	D;7	D;12	E;11	—	—	—	—
Mission 3	A;0	B;0	A;0	C;0	C;0	C;0	C;0	C;0	C;8	D;7	D;12	D;11	D;16	D;15	D;20	E;19

6.2 Parameters setting

There are three key parameters needed to be discussed in the proposed DPMOGA, i.e., the population size NP in each population, the threshold value σ_d^2 in adaptive mutation probability, and the initializing mutation probability P_{m0}. The Taguchi method of design of experiment (DOE) [27,28] is used to determine a set of suitable parameters for the DPMOGA in this section. Combinations of different values of these parameters are shown in Table 4. Experimental evaluation among three mission cases with PS=1.8 is conducted. We run the DPMOGA 20 times independently for each mission case with the maximum number of evaluation Q=10 000. The average response variable (ARV) values are the following average deviation values:

$$\mathrm{ARV} = \sum_{i=1}^{3} \sum_{j=1}^{20} \frac{C_{ij} - LB_i}{LB_i} / (20 \times 3) \tag{23}$$

where C_{ij} is the jth makespan of the ith mission case obtained by the DPMOGA, LB_i is the lowest makespan of the ith mission case obtained by the DPMOGA.

Table 4 Combinations of parameter values

Parameter	Level				
	1	2	3	4	5
NP	30	60	90	120	150
σ_d^2	0.01	0.05	0.09	0.13	0.17
P_{m0}	0.005	0.05	0.1	0.2	0.3

The average ARV of each level for parameters is shown in Table 5, where Delta is the range of average ARV for each parameter, and Rank is the significance rank of each parameter.

According to Table 5, it can be seen that NP is the most significant parameter among the three parameters. The best combination of parameter values are NP=60, $\sigma_d^2 = 0.13$, and $P_{m0} = 0.2$.

Table 5 Average ARV of orthogonal experiments

Level	NP	σ_d^2	P_{m0}
1	0.012 8	0.011 58	0.012 47
2	0.009 2	0.010 61	0.012 05
3	0.014 27	0.013 84	0.011 76
4	0.011 9	0.010 16	0.009 72
5	0.011 29	0.013 28	0.013 47
Delta	0.005 07	0.003 68	0.003 74
Rank	1	3	2

6.3 Comparisons with existing algorithms and discussion

To evaluate the performance of the proposed DPMOGA, experiments comparison among different mission cases is conducted. Four different algorithms for RCPSPs, i.e., the efficient genetic algorithm (EGA) [17], IPSO [13], the multi-modal genetic algorithm (MMGA) [16], and HEDA [14], are used to make a comparison with DPMOGA. All settings for the compared algorithms are using the same

settings proposed in the previous studies.

After 20 independent runs for each algorithm with the maximum number of evaluation Q=10 000, the results are shown in Table 6. The performance of each algorithm is measured by the average makespan (Avg.), the best makespan (Best.), and the variance of 20 results of each algorithm (Var.). Furthermore, according to [29−32], the minLTF priority rule or the minSLK priority rule performs better while solving the RCPSP, so comparisons with priority rules are also conducted. The evolutionary convergence curves of the five algorithms for three missions with two *PS* levels are shown in Fig. 6. In Table 6, for all cases and *PS* levels, the average makespan and the best makespan obtained by DPMOGA are better than that obtained by other algorithms, and except Mission 2 with *PS*=1.8, the variance of results obtained by DPMOGA is also better than that obtained by other algorithms. From the prospective of average makespan in all the mission cases with *PS*=1.8, we can see that as the number of aircraft increases, the difference between the average makespan obtained by DPMOGA and that obtained by other algorithms is increasing. The heuristics using PSGS with the minLTF priority rule and the minSLK priority rule perform poorly, but the calculation speed is fast. Also, the value of *PS* has a strong impact on the makespan. When *PS* is larger, the configured number of personnel in each mission case increases, so the best and the average makespans obtained decrease obviously.

Table 6 Results of experiments

Mission case	*PS*	Performance measurement	Algorithm					Priority rule	
			DPMOGA	EGA	IPSO	HEDA	MMGA	minLFT	minSLK
Mission 1	1.8	Avg.	**75.7**	76.215	76.615	76.345	79.115	93.8	87.3
		Best.	**75.2**	75.7	75.3	75.9	77.4		
		Var.	**0.0758**	0.0824	2.2529	0.25	0.5761		
	3.2	Avg.	**57.565**	58.565	58.79	58.805	59.195	65	63.5
		Best.	**56.8**	58	58	58.5	58.6		
		Var.	**0.055**	0.0971	0.3052	0.0373	0.0637		
Mission 2	1.8	Avg.	**74.695**	75.825	76.6	78.53	79.445	95.6	87.7
		Best.	**74.0**	75.2	74.6	77.3	77.8		
		Var.	0.1689	**0.1472**	2.5274	0.398	0.4847		
	3.2	Avg.	**57.41**	58.3867	59.335	59.085	59.2	63.1	68.8
		Best.	**57.1**	57.4	58.2	58.6	58.7		
		Var.	**0.0378**	0.1998	0.434	0.054	0.0674		
Mission 3	1.8	Avg.	**74.35**	77.72	78.155	80.36	80.83	87.6	83.4
		Best.	**74**	76.1	75.5	79	79.3		
		Var.	**0.0553**	0.5164	2.5626	0.3099	0.8843		
	3.2	Avg.	**61.805**	62.435	63.58	62.815	62.45	66.1	67.4
		Best.	**61.6**	62	62	61.9	61.7		
		Var.	**0.0268**	0.1645	2.0312	0.1413	0.0521		

As for the four compared algorithms, their performance is different in different cases. For EGA, it outperforms the other three compared algorithms from the perspective of the average makespan in Table 6. The variance of results obtained by EGA is relatively small, and especially in all the mission cases with *PS*=1.8, EGA gets the smallest variance in the four compared algorithms. The best makespan obtained by EGA is usually better than that obtained by MMGA and HEDA. In the EGA, another two genes are added into the AL encodings in order to select the SGS and justification direction. Similar to our DPMOGA, this algorithm can adaptively select a better performing SGS and justification direction during the iteration process to generate a scheduling plan. For IPSO, the best makespan obtained is better than that obtained by other three compared algorithms in most mission cases. However, it can be seen in Table 6 that the variance obtained by IPSO is the largest. Also, this algorithm no longer converges after a certain number of iterations as shown in Fig. 6. These phenomena show that the IPSO has poor stability and is easy to fall into the local optimal solution. For HEDA, this algorithm performs moderately but better than MMGA in most mission cases from the perspective of three performance measurements. For MMGA, it usually performs the worst in the four compared algorithms, whether from the perspective of the average makespan or the perspective of the best makespan. We can also find that the convergence speed of the MMGA is slower from Fig. 6. However, MMGA obtains the lowest best makespan in Mission 3 with *PS*=3.2. Another phenomenon (also appears on HEDA) is that, when the value of *PS* decreases, the variance of results obtained by MMGA becomes larger. As the *PS* value decreases, the number of personnel decreases, and the optimization space becomes smaller, so it is difficult for MMGA to find a better solution within a limited number of iterations.

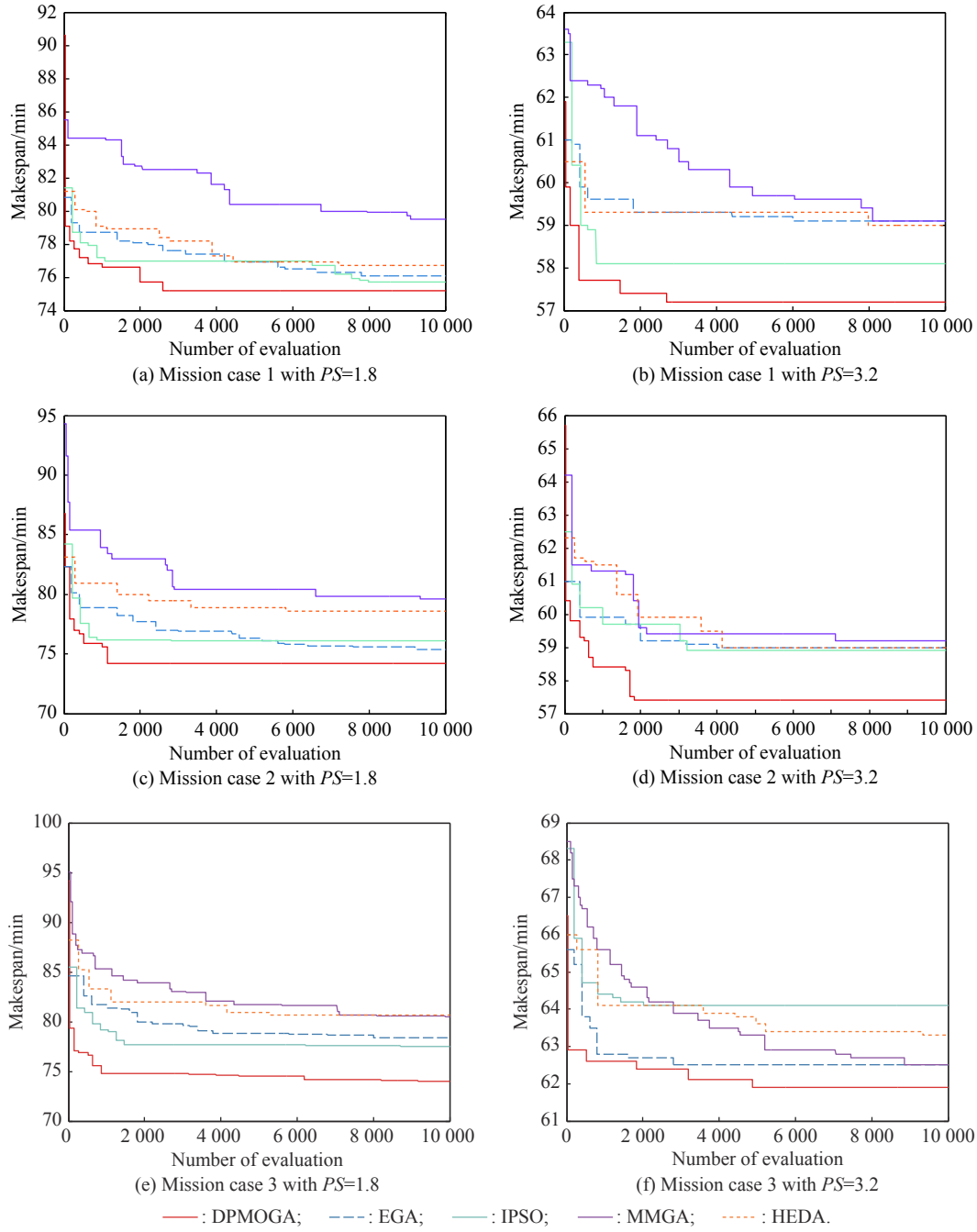


Fig. 6 Evolutionary convergence curves for mission cases

The EGA performs better than the other three compared algorithms. In order to show the improvement of the DPMOGA, the Kruskal-Wallis test for the results obtained by DPMOGA and EGA is carried out. We propose the null hypothesis H_0: results obtained by DPMOGA and EGA come from the same sample. The result of the hypothesis test is shown in Table 7. We can see that the p-values under different test conditions are all extremely small. Therefore, we have sufficient certainty to reject the null hypothesis.

Table 7 Results of the Kruskal-Wallis test

p-value	Mission 1		Mission 2		Mission 3	
	PS=1.8	PS=3.2	PS=1.8	PS=3.2	PS=1.8	PS=3.2
p	3.23×10^{-5}	5.59×10^{-8}	1.12×10^{-7}	3.45×10^{-7}	6.17×10^{-8}	7.62×10^{-7}

7. Conclusion and future work

In this paper, the scheduling problem of flight deck operations for the pre-flight preparation stage is studied. The MAISPTT is considered as a resource-constrained multi-project scheduling problem with transfer times. In view of the NP-hard nature of MAISPTT, the DPMOGA is proposed. In the simulation section, the best combination of parameter values are determined by the Taguchi method. To evaluate the performance of the proposed DPMOGA, simulation experiments among different mission cases with different personnel strengths are conducted. Simulation experiment results show that the DPMOGA outperforms some other state-of-the-art algorithms.

In the future, three main tasks require more in-depth research. Firstly, as we can see in Section 6, the value of *PS* has a strong impact on the makespan. This inspires us to analyze the impact of resource composition, and then study the joint optimization method for scheduling and resource configuration. Secondly, more efficient optimization algorithms for FDOSP need to be studied. Thirdly, many uncertain factors may influence the operation scheduling, so the dynamic scheduling method and the robust scheduling method also need to be considered.

References

[1] LIU J, HAN W, LI J, et al. Integration design of sortie scheduling for carrier aircraft based on hybrid flexible flow-shop. IEEE Systems Journal, 2020, 14(1): 1503–1511.

[2] LIU J, HAN W, WANG X W, et al. Research on cooperative trajectory planning and tracking problem for multiple carrier aircraft on the deck. IEEE Systems Journal, 2020, 14(2): 3027–3038.

[3] WANG X W, LIU J, SU X C, et al. A review on carrier aircraft dispatch path planning and control on deck. Chinese Journal of Aeronautics, 2020, 33(12): 3039–3057.

[4] SHI W W, HAN W, SI W C. Optimization of direct flight line maintenance process for multi-carrier planes. Computer Engineering and Design, 2013, 34(12): 4214–4219. (in Chinese)

[5] YU L F, ZHU C, SHI J M, et al. An extended flexible job shop scheduling model for flight deck scheduling with priority, parallel operations, and sequence flexibility. Scientific Programming, 2017, 2017: 2463252.

[6] CUI R W, HAN W, SU X C, et al. A multi-objective hyper heuristic framework for integrated optimization of carrier-based aircraft flight deck operations scheduling and resource configuration. Aerospace Science and Technology, 2020, 107: 106346.

[7] JIANG T T, SU X C, HAN W. Optimization of support scheduling on deck of carrier aircraft based on improved differential evolution algorithm. Proc. of the 3rd International Conference on Control Science and Systems Engineering, 2017. DOI: 10.1109/CCSSE.2017.8087910.

[8] SU X C, HAN W, XIAO W, et al. Pit-stop support scheduling on deck of carrier plane based on memetic algorithm. Systems Engineering and Electronics, 2016, 38(10): 2303–2309. (in Chinese)

[9] YUAN P L, HAN W, SU X C, et al. A dynamic scheduling method for carrier aircraft support operation under uncertain conditions based on rolling horizon strategy. Applied Sciences, 2018. DOI: 10.3390/app8091546.

[10] SU X C, HAN W, WU Y, et al. A proactive robust scheduling method for aircraft carrier flight deck operations with stochastic durations. Complexity, 2018. DOI: 10.1155/2018/6932985.

[11] SU X C, HAN W, WU Y, et al. A robust scheduling optimization method for flight deck operations of aircraft carrier with ternary interval durations. IEEE Access, 2018, 6: 69918–69936.

[12] ELSAYED S, SARKER R, RAY T, et al. Consolidated optimization algorithm for resource-constrained project scheduling problems. Information Sciences, 2017, 418: 346–362.

[13] JIA Q, SEO Y. An improved particle swarm optimization for the resource-constrained project scheduling problem. International Journal of Advanced Manufacturing Technology, 2013, 67(9−12): 2627–2638.

[14] WANG L, FANG C. A hybrid estimation of distribution algorithm for solving the resource-constrained project scheduling problem. Expert Systems with Applications, 2012, 39(3): 2451–2460.

[15] CHALESHTARTI A S, SHADROKH S, KHAKIFIROOZ M, et al. A hybrid genetic and Lagrangian relaxation algorithm for resource-constrained project scheduling under nonrenewable resources. Applied Soft Computing, 2020, 94: 106482.

[16] PEREZ E, POSADA M, LORENZANA A. Taking advantage of solving the resource constrained multi-project scheduling problems using multi-modal genetic algorithms. Soft Computing, 2015, 20(5): 1–18.

[17] KADRI R L, BOCTOR F F. An efficient genetic algorithm to solve the resource-constrained project scheduling problem with transfer times: the single mode case. European Journal of Operational Research, 2017, 265(2): 454–462.

[18] DEBELS D, VANHOUCKE M. A decomposition-based genetic algorithm for the resource-constrained project-scheduling problem. Operations Research, 2007, 55(3): 457–469.

[19] ZHANG X J, MA S, CHEN S L. Healthcare service configuration based on project scheduling. Advanced Engineering Informatics, 2020, 43: 101039.

[20] HAN W, SU X C, CHEN J F. Integrated maintenance support scheduling method of multi-carrier aircraft. Systems Engineering and Electronics, 2015, 37(4): 809–816. (in Chinese)

[21] HARTMANN S, KOLISCH R. Experimental evaluation of state-of-the-art heuristics for the resource-constrained project scheduling problem. European Journal of Operational Research, 2000, 127(2): 394–407.

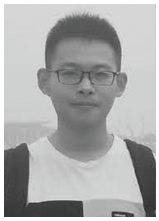
[22] KOLISCH R, HARTMANN S. Heuristic algorithms for the resource-constrained project scheduling problem: classification and computational analysis. http:// hdl.handle.net/10419/147576.

[23] DEBELS D, REYCK B D, LEUS R, et al. A hybrid scatter search/electromagnetism meta-heuristic for project scheduling. European Journal of Operational Research, 2006, 169(2): 638–653.

[24] WANG J Q, CHENG Z W, ZHANG P L, et al. Research on improvement of real-coded genetic algorithm for solving constrained optimization problems. Control and Decision, 2019, 34(5): 44–53. (in Chinese)

[25] COELHO J, VANHOUCKE M. Going to the core of hard resource-constrained project scheduling instances. Computers & Operations Research, 2020, 121: 104976.
[26] KOLISCH R. Serial and parallel resource-constrained project scheduling methods revisited: theory and computation. European Journal of Operational Research, 1996, 90(2): 320–333.
[27] RADY RAZ N, AKBARZADEH-T M R, AKBARZADEH A. Experiment-based affect heuristic using fuzzy rules and Taguchi statistical method for tuning complex systems. Expert Systems with Applications, 2021, 172: 114638.
[28] HOSEINPOUR-LONBAR M, ALAVI M Z, PALASSI M. Selection of asphalt mix with optimal fracture properties at intermediate temperature using Taguchi method for design of experiment. Construction and Building Materials, 2020, 262: 120601.
[29] KRUGER D, SCHOLL A. A heuristic solution framework for the resource constrained (multi-) project scheduling problem with sequence-dependent transfer times. European Journal of Operational Research, 2009, 197(2): 492–508.
[30] GUO W, VANHOUCKE M, COELHO J, et al. Automatic detection of the best performing priority rule for the resource-constrained project scheduling problem. Expert Systems with Applications, 2020, 167: 114116.
[31] CHAKRABORTTY R K, RAHMAN H F, RYAN M J. Efficient priority rules for project scheduling under dynamic environments: a heuristic approach. Computers & Industrial Engineering, 2020, 140: 106287.
[32] CHEN H, DING G, ZHANG J, et al. Research on priority rules for the stochastic resource constrained multi-project scheduling problem with new project arrival. Computers & Industrial Engineering, 2019, 137: 106060.

Biographies



CUI Rongwei was born in 1996. He received his B.S. degree in mechanical engineering from Naval Aviation University, Yantai, China, in 2019. He is currently working toward his master's degree in Naval Aviation University. His research interests include flight deck operations scheduling and intelligent computation.
E-mail: cuirongwei126@163.com

HAN Wei was born in 1970. He received his B.S. degree in aircraft and engine engineering and M.S. degree in aeronautical and astronautical science and technology from Naval Aviation University, Yantai, China, in 1992 and 1996, respectively, and Ph.D. degree in solid mechanics from Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2003. He is a professor in Naval Aviation University. His research interests include operations scheduling and aircraft dynamics.
E-mail: Hanwei70cn@163.com

SU Xichao was born in 1989. He received his B.S. degree in aircraft system and engineering and Ph.D. degree in aeronautical and astronautical science and technology from Naval Aviation University, Yantai, China, in 2012 and 2018, respectively. He is a lecturer in Naval Aviation University. His research interests are flight deck operations scheduling and intelligent computation.
E-mail: suxich@126.com

LIANG Hongyu was born in 1995. He received his B.S. degree in mechanical engineering and M.S. degree in aeronautical and astronautical science and technology from Naval Aviation University, Yantai, China, in 2017 and 2019, respectively. He is an assistant engineer in Unit 91404 of the PLA. His research interests are flight deck operations scheduling and intelligent computation.
E-mail: yu675878@163.com

LI Zhengyang was born in 1994. He received his B.S. degree in water resources and hydropower engineering from Wuhan University, Wuhan, China, in 2017, and M.S. degree in aviation engineering from Naval Aviation University, Yantai, China, in 2019. He is an assistant engineer in Unit 91404 of the PLA. His research interests are flight deck operations scheduling and intelligent computation.
E-mail: lizhengyang1021@163.com