Mathematical Biosciences and Engineering

http://www.aimspress.com/journal/mbe

MBE, 20(1): 1460–1487.
DOI: 10.3934/mbe.2023066
Received: 29 August 2022
Revised: 19 October 2022
Accepted: 21 October 2022
Published: 31 October 2022

Research article

Hybrid multi-objective metaheuristic algorithms for solving airline crew rostering problem with qualification and language

Bin Deng[1,*], **Ran Ding**[1], **Jingfeng Li**[2], **Junfeng Huang**[2], **Kaiyi Tang**[2] **and Weidong Li**[1]

[1] School of Mathematics and Statistics, Yunnan University, Kunming 650000, China
[2] China Eastern Yunnan Airlines, Kunming 650200, China

* **Correspondence:** Email: dengbin96@126.com.

Abstract: In order to cope with the rapid growth of flights and limited crew members, the rational allocation of crew members is a strategy to greatly alleviate scarcity. However, if there is no appropriate allocation plan, some flights may be canceled because there is no pilot in the scheduling period. In this paper, we solved an airline crew rostering problem (CRP). We model the CRP as an integer programming model with multiple constraints and objectives. In this model, the schedule of pilots takes into account qualification restrictions and language restrictions, while maximizing the fairness and satisfaction of pilots. We propose the design of two hybrid metaheuristic algorithms based on a genetic algorithm, variable neighborhood search algorithm and the Aquila optimizer to face the trade-off between fairness and crew satisfaction. The simulation results show that our approach preserves the fairness of the system and maximizes the fairness at the cost of crew satisfaction.

Keywords: airline crew rostering problem; multi-objective optimization; genetic algorithm; variable neighborhood search algorithm; Aquila optimizer

1. Introduction

The aviation industry is an important industry in the current market. As a modern mode of transportation, air transport plays an important role in economic activities. Since the founding of New China, China has become the second largest civil aviation market in the world. In 2021, the total transportation turnover of the whole industry will be 85.675 billion ton- kilometer, an increase of 7.3% over the previous year. The total passenger traffic volume of the whole industry reached 440.557 million passengers, an increase of 5.5% over the previous year [1].

At present, most airlines' crew scheduling relies on the staff's experience to do it manually. Although it is practical, there are obvious shortcomings and the probability of error is not low. The essence of the problem is an assignment problem. Many institutions and companies have carried out various degrees of discussion and research on related issues. This is an non-deterministic polynomia hard (NP-hard) problem, and it has always been a hot topic of research by scholars at home and abroad. The crew rostering problem (CRP) is a complex and huge workload. The crew resource is one of the most precious resources in aviation operation, as it also determines the importance and necessity of crew scheduling in aviation operation management. Figure 1 shows the number of captains and copilots of some Chinese airlines in 2019. Due to the large scale of flights involved and the need to strictly abide by complex working rules, in order to reduce the difficulty of solving the crew scheduling, the crew scheduling is usually divided into two parts: the crew pairing problem (CPP) and crew rostering problem (CRP). We must solve the CPP to get a set of flight pairings with excellent cost and quality performance. The CRP is to assign these flight pairings to the crew. In most existing studies, the goal of the CRP is set to minimize airline costs [2]. However, with the development of the aviation industry, the cost of airlines is no longer the primary priority of crew scheduling. According to statistics, it takes 5 million yuan for an airline to train a captain. The loss of a pilot is a huge loss for the airline; especially, a highly qualified pilot is a valuable asset for the airline. Therefore, fairness and satisfaction are key factors to be taken into account by the rosters when assigning flight pairings [3]. In this paper, we also consider a multi-objective model, including fairness and satisfaction.

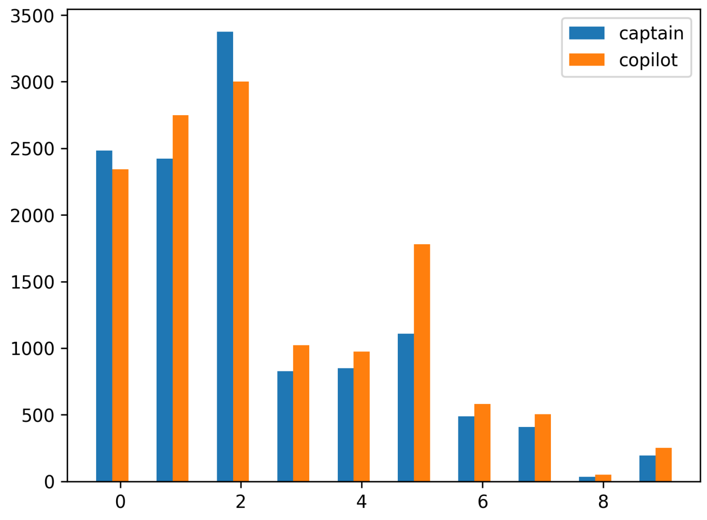


Figure 1. Number of pilots and copilots of airlines in 2019. On the abscissa, from left to right are China Eastern, Air China, China Southern Airlines, Sichuan Airlines, XIAMEN AIR, Hainan Airlines, Spring Airlines, Juneyao Airlines, YTO Cargo Airlines and SF Cargo Airlines.

In China, there are many plateau airports and special airports, especially in Yunnan. Only pilots with these qualifications can fly these airports, so these pilots are more scarce resources for airlines. Therefore, in this paper, we mainly study the CRP with language and qualification constraints, and the CPP will not be considered for the time being. A multi-objective model with the following two objectives has been built for the CRP: 1) maximize fairness and 2) maximize satisfaction. The purpose is to find a set of Pareto optimal solutions among a variety of possible combinations under the various regulations of the Civil Aviation Administration. The main difficulty of this problem is that, with the increase of the number of pilots and flights, the size of the portfolio grows exponentially and becomes difficult to solve. With the increase of the number of pairings and crew, the size of the search space increases dramatically. Therefore, it is impractical if we enumerate each combination when the problem size is large.

The multi-objective model we propose is a discrete optimization model [4]. It is difficult for traditional mathematical methods to yield optimal solutions for multiple objectives at the same time. Therefore, three metaheuristic algorithms are mainly considered in this paper, including a genetic algorithm (GA), the Aquila optimizer (AO) and a variable neighborhood search (VNS) algorithm. However, only using a single algorithm makes it easy to fall into the local optimal solution, so we have designed two hybrid metaheuristic algorithms. One is a hybrid algorithm of a GA and VNS, and the other is a hybrid algorithm of a GA and a AO.

In this paper, we have three contributions:

1) We establish the CRP as a multi-objective model with qualifications and language constraints.

2) We propose two hybrid metaheuristic algorithms to effectively solve the proposed multi-objective CRP.

3) In this study, we tested monthly scheduling problems of different scales. At the same time, we have carried out many simulation experiments to verify the effectiveness of this method. From the experimental results, we can see that our algorithm is effective in large-scale CRPs.

The rest of this paper is structured as follows. In the second section, we review the relevant literature on crew scheduling. In the third section, we establish a mathematical programming model for the CRP. Then, in the fourth section, we introduce two hybrid heuristic algorithms and the design of the coding method. The fifth section analyzes the experimental results of the algorithm; finally, we give the conclusion in the sixth section.

2. Related work

Manual scheduling is time-consuming and laborious for airlines, and it is difficult to control multiple targets. Therefore, in recent years, many scholars have studied crew scheduling optimization in airlines. Next, we will review relevant literature from the perspective of objective function types.

2.1. Single objective model

In many engineering problems, most researchers first consider a single objective model [5, 6], or convert multiple objectives into a single objective model for solution. The same is true for the CRP. For crew scheduling, in most studies, single objectives such as minimizing total cost, maximizing crew satisfaction and balancing working hours have been considered. For example, Beasley et al. [7] considered a problem of assigning K individuals to N tasks with fixed start times and fixed end times. They established a 0-1 integer programming model for this problem, and in order to solve this model, they proposed a tree search algorithm. Later, they found a new lower bound for the crew scheduling problem based on dynamic programming; they then combined this lower bound into the tree search algorithm to solve the problem of random generation between 50 and 500 [8]. In order to solve the crew scheduling problem, Lučić et al. [9] constructed the monthly schedule of the crew using simulated annealing, GA and Tabu search techniques. In order to reduce the overall operating cost of airlines, Maenhout et al. [10] proposed a decentralized search algorithm for airline crew scheduling. Hadianti et al. [11] considered Indonesian airlines. They took the average relative deviation between the total flight time and the ideal flight time as the objective function, and then used the simulated annealing algorithm to solve it. The experimental results show that a satisfactory solution can be obtained in a very short time. Quesnel et al. [12] considered the preferences of the crew and proposed to consider their preferences in the CPP in order to create a pairing that makes the crew more satisfied. At the same time, they used the column generation algorithm to obtain a solution. In order to maximize the satisfaction of the crew, Quesnel et al. [13] proposed a partial pricing scheme based on deep learning. The experimental results show that the solution generated by the branch pricing algorithm can be solved in half the time of the branch pricing algorithm. Recently, Deng [14] proposed an improved honey badger algorithm to solve the models while considering cost and fairness; they achieved good results.

The work mentioned above only considers CRPs, but there are some studies that also consider CPPs and CRPs. Souai et al. [15] proposed a new method to solve the CPP and CRP simultaneously based on a hybrid GA. Saddoune et al. [16] considered an ensemble crew scheduling model and developed a combined column generation algorithm to obtain the solution. Recently, Zeighami et al. [17] proposed a model integrating crew pairing and personalized allocation, and developed an algorithm integrating alternating Lagrangian decomposition, column generation and dynamic constraint aggregation to solve it. Although the integrated model can be optimized globally, it takes a long time to directly solve the integrated model.

2.2. *Multi-objective model*

In fact, multi-objective optimization is also considered in some related fields, including health care routing [18], supply chain management [19], vehicle routing management [20] and flow-shop scheduling [21]. Naturally, we will also consider whether to describe the CRP as a multi-objective problem. In the real world, in addition to paying attention to costs, airlines also pay attention to pilots' fatigue, fairness and other indicators. Therefore, multi-objective models have also been investigated in some studies. Ehrgott et al. [2] considered not only the cost of the solution, but also the robustness of the solution. They described these two objectives as a multi-objective problem and developed a double diagonal optimization framework to generate Pareto schedules for airlines. A multi-meme memetic algorithm improves reliability and flexibility in the real world by Burke et al. [22]. Chutima et al. [23] considered four optimization objectives and solved the crew scheduling problem of a low-cost airline. Zhou et al. [3] proposed a multi-objective ant colony algorithm to optimize the fairness and satisfaction of the crew. Baradaran et al. [24] considered two objectives, where one is to maximize the number of planned vacation days and the other is to minimize the penalty costs associated with violating the minimum and maximum working hours.

In recent years, COVID-19 has affected the development of the aviation industry to a certain extent. However, with the control of COVID-19, the aviation industry has also begun to recover. Therefore, it is a matter of concern to consider the airline CRP from the perspective of pilots. Although people have done a lot of research on the airline CRP, there are few studies that consider the fairness and satisfaction of pilots at the same time. Therefore, in this study, we designed two multi-objective optimization algorithms to solve the problem of airline crew rostering.

3. Preliminaries

In this section, we introduce the input information, constraints and objective function of the CRP. Then we describe a mathematical model considering qualification and language. Before doing these works, we will first explain the terms in the CRP so as to help people without relevant knowledge to understand the article well.

Flight segment: A flight from one airport to another (without a third airport stop in between).

Crew: In this paper, the crew includes only the pilot. In modern civil aviation, there are usually only two types of pilots, namely, the captain and co-pilot.

Deadhead: It refers to the process in which the crew members take an airplane or ground transportation to complete the flight task according to the requirements of the company, but it does not include the transportation to and from the local accommodation.

Duty: Duty consists of connection times between flight segments. The starting time of duty is calculated from the departure time of the first flight performed on the day, and the end time is calculated according to the arrival time of the last flight.

Pairing: Consecutive days of tasks originating from the base and eventually returning to the base, which may include deadhead.

Personal schedule: The work schedule of a crew member over a longer period is linked by a series of flight pairings and other training and vacation arrangements.

Time: Airline operations span time and space. All times are defined in singer same time zone (such as the east eighth time zone), and the time division point of two adjacent days is midnight in the given time zone.

Roster: Each crew member has a schedule within a schedule period, which consists of a series of pairings.

As shown in Table 1, each row in the table represents a flight; it includes the departure airport and departure time, as well as the arrival airport and arrival time. At the same time, Table 2 shows its corresponding flight pairing, where $P1 = [L1, L2, L3, L4]$ and $P2 = [L5, L6, L7, L8]$.

Table 1. Example of flight information.

Leg	Airport Dep	Time Dep	Airport Arr	Time Arr
L1	Airport 1	2020-03-01 09:00	Airport 2	2020-03-01 11:00
L2	Airport 2	2020-03-01 13:00	Airport 1	2020-03-01 15:10
L3	Airport 1	2020-03-02 08:40	Airport 3	2020-03-02 10:34
L4	Airport 3	2020-03-02 11:20	Airport 1	2020-03-02 13:00
L5	Airport 1	2020-03-01 09:00	Airport 2	2020-03-01 11:00
L6	Airport 2	2020-03-01 13:00	Airport 1	2020-03-01 15:20
L7	Airport 1	2020-03-02 09:00	Airport 3	2020-03-02 11:00
L8	Airport 3	2020-03-02 13:00	Airport 1	2020-03-02 15:20

Table 2. Example of pairing information.

Pairing No.	Starting time	End time	Flight time	Duty time	Language	Qualification
P1	2020-03-01 09:00	2020-03-02 13:00	464 min	570 min	Chinese	H
P2	2020-03-01 09:00	2020-03-02 15:20	520 min	760 min	Chinese	S

Note: H: High plateau airport qualification; S: Special airport qualification.

3.1. *Input information*

Whether manual or automated, we rely on two main types of information: pairing and crew. In the actual scheduling system, the information from the aviation planning stage is usually recorded, and the information from the flight pairing stage and the information of the crew are also recorded. For the crew, as in Table 3, it usually includes the flight time of the current month, the flight time of the year, rank, qualification, language, etc. For the pairings, as in Table 2, they usually include flight time, duty time, start time, end time, minimum required qualifications, required language, etc. Therefore, we will use this known information to find a near-optimal personal schedule for the CRP.

Table 3. Example of crew information.

Crew No.	Rank	Monthly flight time	Annual flight time	Language	Qualification
n1	A1	37.23 h	110.45 h	English and Chinese	H
n2	C1	32.56 h	112.32 h	Chinese	S

Note: H: High plateau airport qualification; S: Special airport qualification.

3.2. *Constraints and objectives*

In reality, the CRP is very complicated; airlines in different countries may have different restrictions, and even different airlines in the same country have different restrictions. In this paper, only the crew scheduling problem of Chinese airlines is studied. In order to simplify the process, we will not consider all of the constraints of the airline, but only some very important constraints. For some other constraints, if necessary, you can directly limit the corresponding restrictions in the code.

Constraints

1) Number of crew constraints: Each pairing must be assigned a given number of crew members. For example, a pairing requires a minimum of two crew members, which is not feasible if only one crew member is assigned.

2) Rank constraints: The Captain's and First Officer's ranks must be compatible to be assigned to the pairing together.

3) Flight time constraints: The crew's monthly and annual flying hours must be within the specified limits.

4) Language constraints: At least one crew member flying the same pairing speaks the local language.

5) Qualification constraints: Each crew member must have the qualifications required by the take-off and landing airports.

6) Rest time constraints: The prescribed rest time must be satisfied between the two pairings assigned to each pilot.

7) Flight coverage constraints: All flights must be fully allocated.

Objectives

Optimizing preference: Before the CRP, the pairing was already formed. Therefore, all crew members can express their satisfaction with the pairing on the portal. In our work, the average preference of all pilots is maximized. We set five levels of crew satisfaction, as shown in Table 4.

Table 4. Table of satisfaction score.

Satisfaction	Very dissatisfied	Dissatisfied	Generally	Satisfied	Very satisfied
Points	1	2	3	4	5

Optimizing fairness: Workload balancing is an important indicator for achieving crew fairness. An equitable schedule can lead to increased crew motivation.

3.3. Mathematical model

To build the mathematical model, we first define the main sets, parameters and decision variables, as shown in Table 5. In Table 5, although we describe each crew member's individual schedule as Set. But we should know that, as a legal individual scheduling plan, they each should satisfy the qualification constraints, language constraints and grade constraints, etc.

Table 5. Summary of notations.

Type of notation	Notation	Description
Set	P	Set of all pairings.
	M	Set of crew members.
	R_m	The set of all legal individual schedules of crew member m.
Parameter	c_{rm}	Total cost of crew member m to execute the schedule $r \in R_m$.
	c_p	The cost of pairing p.
	a_{rp}	$a_{rp} = 1$ if pairing p is in the schedule r, 0 otherwise.
	$b_{m_1m_2}$	$b_{m_1m_2} = 1$ if crew members m_1 and m_2 can be matched, 0 otherwise.
	s_{rm}	Crew m's satisfaction with the schedule r.
	l_{pm}	$l_{pm} = 1$ if crew member m satisfies the language requirements of pairing p, 0 otherwise.
	q_{pm}	$q_{pm} = 1$ if crew member m satisfies the qualification requirements of pairing p, 0 otherwise.
Variable	x_{rm}	$x_{rm} = 1$ if the crew m performs the schedule r, 0 otherwise.

Here, we consider two main objectives of fairness and satisfaction to build a multi-objective optimization model for the CRP. The mathematical model can be described as follows:

$$Min \qquad f_1 = \frac{\sum_{m\in M}\sum_{r\in R_m}(c_{rm} - \bar{c})x_{rm}}{\mid M \mid} \tag{3.1}$$

$$Max \qquad f_2 = \frac{\sum_{m\in M}\sum_{r\in R_m} s_{rm}x_{rm}}{\mid M \mid} \tag{3.2}$$

subject to:

$$\sum_{m_1 \in M} \sum_{m_2 \in M: m_2 \neq m_1} \sum_{r_1 \in R_{m_1}} \sum_{r_2 \in R_{m_2}} b_{m_1 m_2}(a_{r_1 p} x_{r_1 m_1} + a_{r_2 p} x_{r_2 m_2}) = 2 \qquad \forall p \in P \tag{3.3}$$

$$\sum_{m \in M} \sum_{r \in R_m} q_{pm} a_{rp} x_{rm} = 1 \qquad \forall p \in P \tag{3.4}$$

$$\sum_{m_1 \in M} \sum_{m_2 \in M: m_2 \neq m_1} \sum_{r_1 \in R_{m_1}} \sum_{r_2 \in R_{m_2}} (l_{pm_1} a_{rp} x_{rm_1} + l_{pm_2} a_{rp} x_{rm_2}) \geq 1 \qquad \forall p \in P \tag{3.5}$$

$$\sum_{r \in R_m} x_{rm} = 1 \qquad \forall m \in M \tag{3.6}$$

$$x_{rm} \in \{0, 1\} \qquad \forall r \in R_m, \forall m \in M \tag{3.7}$$

Objective function (3.1) optimizes fairness among crew members by minimizing the sum of the deviations of the selected schedule cost value from the average cost. Here we take the form of mean absolute deviation. In fact, the average cost is actually a constant, and it is calculated as follows: $\overline{c} = \sum_{p \in P} c_p / \mid M \mid$. Objective function (3.2) maximizes average satisfaction among the crew members. Constraint (3.3) ensures that each ring is executed by two crew members, while guaranteeing that the two crew members can be matched in rank. Constraint (3.4) ensures that the qualifications of each crew member meet the requirements of the pairing. Constraint (3.5) ensures that at least one crew member meets the language requirements of the pairing. Constraint (3.6) ensures that each crew member selects a schedule. The binary conditions are defined by Constraint (3.7).

4. Solution approach

In the real world, there are often a large number of large-scale optimization problems. For this type of problem, either it is often difficult to solve with exact algorithms, or it takes a lot of time to solve. Therefore, in recent years, many metaheuristic algorithms have been proposed, including particle swarm optimization, a GA, honey badger algorithms, etc., and successfully applied them to different problems [14, 25, 26]. But a single algorithm may not perform well on some problems. Therefore, in this regard, hybrid optimization algorithms are of great significance for solving real-world large-scale optimization problems. In this paper, we propose two different hybrid methods to solve this problem. The first method was constructed by using a hybrid Non-dominated sorting genetic algorithm II (NSGA-II) and VNS algorithm, which we call GA-VNS. The second algorithm was constructed using the newly proposed AO [27] and a GA, called AOGA.

4.1. Encoding and decoding schemes

For the input information of the question, the pilots are indexed by number. Therefore, in order to solve the proposed CRP model, we first need to digitally encode the pilot's information. A reasonable coding scheme can simplify the problem. In our problem, we put all the pilots in a list; when the list is fixed, the order of the pilots is fixed. We code the pilots in the order they appear in the list, i.e., the pilot in the first position in the list will be coded it as 1, and the pilot in the second position will be coded it as 2, as shown in Figure 2. In Figure 2, the upper part is the input information of the airline, and the middle part is the information encoded for the pilot. Once we have coded the pilot's information, we also need to code the solution. Since each flight pairing requires two pilots, we describe the solution as an array. Each row in the array represents a flight ring, while the corresponding first column represents the captain and the second column represents the first officer, as shown at the bottom of Figure 2. When we decode the solution obtained by the algorithm, we only need to take the number of the corresponding position from the list of pilots and flight pairings.

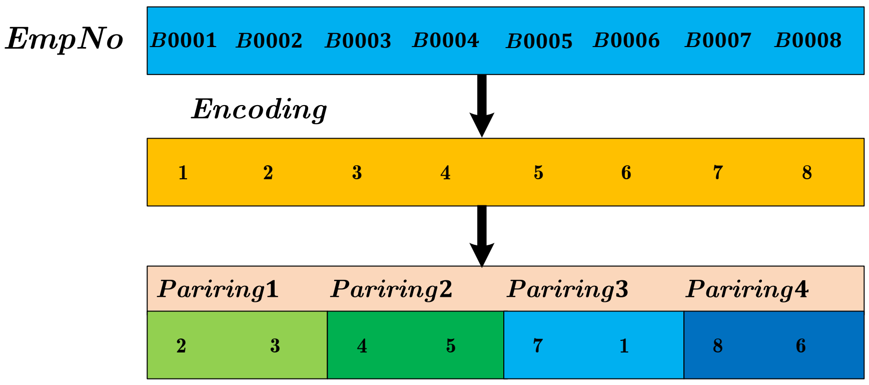

EmpNo
B0001 B0002 B0003 B0004 B0005 B0006 B0007 B0008
Encoding
1 2 3 4 5 6 7 8
Pariring1 Pariring2 Pariring3 Pariring4
2 3 4 5 7 1 8 6

Figure 2. Encoding.

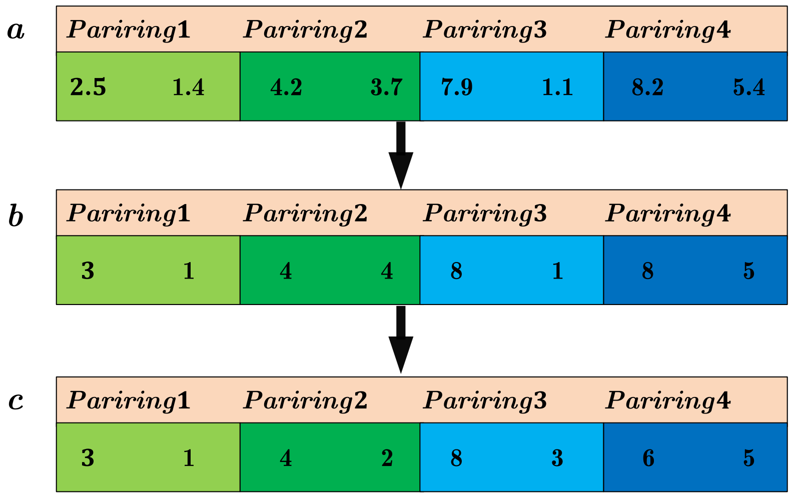

a
Pariring1 Pariring2 Pariring3 Pariring4
2.5 1.4 4.2 3.7 7.9 1.1 8.2 5.4
b
Pariring1 Pariring2 Pariring3 Pariring4
3 1 4 4 8 1 8 5
c
Pariring1 Pariring2 Pariring3 Pariring4
3 1 4 2 8 3 6 5

Figure 3. RK technique.

In general, optimization algorithms are often used in continuous optimization. Therefore, in order to be able to search in the feasible space, we will use the random-key (RK) technique proposed in [28]. This technique is divided into two stages, where the first stage uses an algorithm to generate a continuous solution, and the second stage parses this continuous solution into a feasible solution. This technique is often used because it can apply continuous optimization algorithms to discrete problems. For example, suppose we need to allocate four flight pairings on the same day. Figure 3a is a real solution generated by the algorithm. We then generate an integer solution by rounding, as shown in Figure 3b. An integer solution is obtained by rounding, but, obviously, this integer solution is not feasible. Therefore, we need to further obtain a feasible solution to the problem, as shown in Figure 3c.

4.2. Multi-objective optimization

After defining the search space and coding method, we also need to evaluate the pros and cons of each solution. In our proposed CRP model, there are two conflicting objectives. Therefore, it is impossible for us to optimize both objectives at the same time, but we need to make a trade-off between the two objectives. In this case, instead of a single solution, we get a set of Pareto solutions. For every Pareto solution, we cannot optimize one objective without degrading the other. Therefore, in this work, our goal is to find a set of Pareto solutions. Suppose we get two Solutions A and B. Their corresponding objective functions are $(f_1(A), f_2(A))$ and $(f_1(B), f_2(B))$, respectively. Solution A dominates Solution B if $f_1(A) \leq f_1(B)$, $f_2(A) \geq f_2(B)$ and $(f_1(A), f_2(A)) \neq (f_1(B), f_2(B))$. We divide the population set into different Pareto sets by crowding distance. Obviously, the first Pareto front is the non-dominant solution. In a metaheuristic algorithm, the solution set for each iteration is divided into different Pareto sets. Next, we will propose a hybrid metaheuristic algorithm.

4.3. Non-dominated sorting genetic algorithm II

The NSGA-II is a popular algorithm for solving multi-objective optimization problems. The basic idea of the NSGA-II is to rank the population through the non-dominated sorting of the population, calculate the crowding distance of the population of individuals to maintain the diversity of the population and obtain the non-dominated solution when the termination condition is reached. The NSGA-II randomly selects two individuals from the parent population as the father and mother. Then it performs the crossover operation with the probability P_c, and performs a mutation operation with the probability P_m.

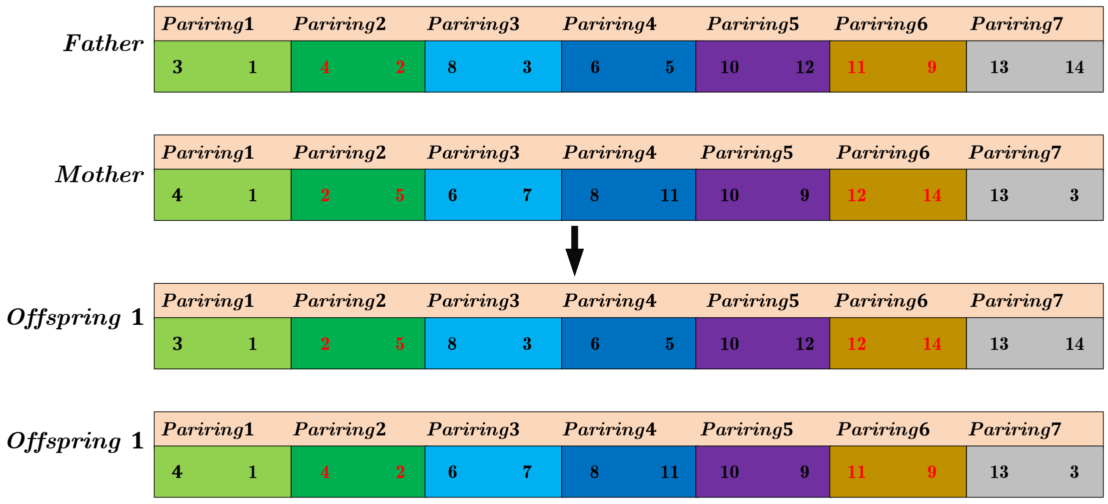


Figure 4. Crossover operation.

Crossover operation: First, we use a random function $RC = randi(1, \mid P \mid /2)$ to generate the number of cross positions, where $\mid P \mid$ is the number of flight pairings. Then, the RC cross positions are generated through random functions $randi(1, | P |)$. In Figure 4, we show an example of a crossover operation. In Figure 4, the number of intersecting positions is 2; therefore, we need to randomly generate two crossover positions. Here, we produce two crossover positions 2 and 6. Then, we swap the second and sixth positions of Father and Mother to get two offsprings 1 and 2.

Mutation operation: As shown in Figure 5, we use the randomly generated mutation position 2. Then, we randomly turn that position into an element in the feasible space.

We merge parent and child, using non-dominant sorting and crowding distance to produce a new population of the size of the initial population. The flow chart of the NSGA-II is shown in Figure 6.

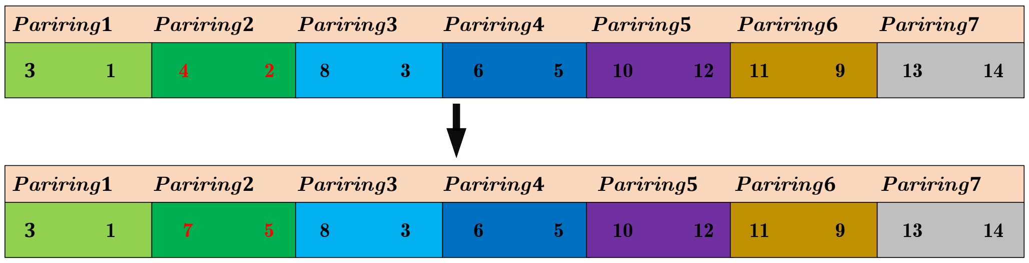


Figure 5. Mutation operation.

4.4. Variable neighborhood search algorithm

The VNS algorithm is one of the most popular local search algorithms [29], and it is often used to solve large-scale optimization problems. The VNS algorithm is an improved local search algorithm. It utilizes the neighborhood structure formed by different actions for alternate search and achieves a good balance between concentration and evacuation. The VNS algorithm relies on the following facts: 1) A local optimal solution for one neighborhood structure is not necessarily a local optimal solution for another neighborhood structure; 2) The global optimum is the local optimum for all possible neighborhoods. The VNS algorithm starts from a set of initial solutions and uses N_{max} neighborhood structures to find a better solution than the current one. Therefore, the effect of the VNS algorithm mainly depends on the design of the neighborhood structure. Therefore, we can reasonably design the domain structure to be embedded in other algorithms to improve the solution effect of the algorithm. In this work, three types of neighborhood structures were mainly designed for the CRP, as follows:

Pilot exchange: In a solution X, randomly select two pilots to exchange, as shown in Figure 7 (Pilot exchange).

Insert: In a solution X, a pilot is randomly selected from the unassigned pilots and inserted into a flight pairing at random, as shown in Figure 7 (Insert).

Pairing exchange: In a solution X, randomly select two pairings to exchange, as shown in Figure 7 (Pairing exchange).

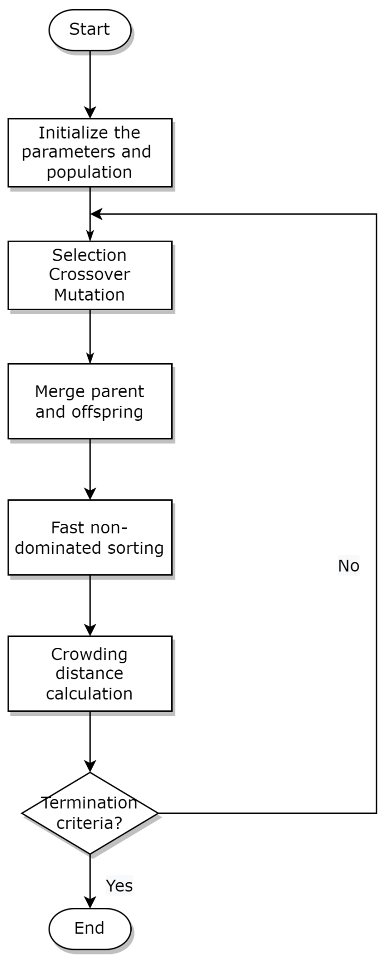

Start
Initialize the parameters and population
Selection
Crossover
Mutation
Merge parent and offspring
Fast non-dominated sorting
Crowding distance calculation
Termination criteria?
No
Yes
End

Figure 6. Flow chart of the NSGA-II.

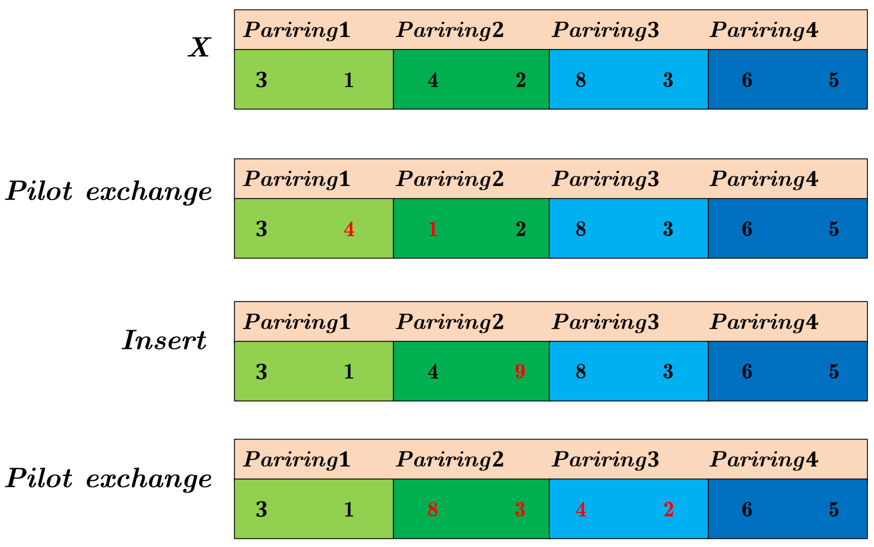

X
Pariring1 Pariring2 Pariring3 Pariring4
3 1 4 2 8 3 6 5
Pilot exchange
Pariring1 Pariring2 Pariring3 Pariring4
3 4 1 2 8 3 6 5
Insert
Pariring1 Pariring2 Pariring3 Pariring4
3 1 4 9 8 3 6 5
Pilot exchange
Pariring1 Pariring2 Pariring3 Pariring4
3 1 8 3 4 2 6 5

Figure 7. Neighborhood structure.

4.5. Aquila optimizer

The AO is an optimization algorithm proposed by Abualigah et al. in 2021 based on the hunting behavior of Aquila [27]. Due to the ability of the algorithm's advanced evolutionary strategy to find the global optimum, it has been widely used in a variety of optimization problems [30,31]. Like other metaheuristics, the AO starts from an initial population. In nature, there are four types of hunting methods. Aquila bend vertically, soar high in the sky and select their search space. They conduct high-altitude exploration in a forked search space. The Aquila flies at low altitude and slowly descends to attack in the convergent search space. And, Aquila walk to catch prey.

4.5.1. Step 1: Expanded exploration

In the first hunting style, Aquila soar through the sky to determine the range of their prey. This hunting behavior can be expressed mathematically by the following formula:

$$X_1(t+1) = X_{best}(t) * (1 - \frac{t}{Max\_iter}) + (X_M(t) - X_{best}(t) * r),$$

where $X_1(t+1)$ is the solution for the $t+1$-th iteration, which is produced by the Aquila's first method. $X_{best}(t)$ is the best solution so far. This equation $(1 - \frac{t}{T})$ controls the hunting range of the Aquila through the number of iterations. r is a random value between 0 and 1. $X_M(t)$ represents the average of all solutions in the t-th iteration. t and Max_iter are the current iteration value and the maximum iteration value, respectively.

4.5.2. Step 2: Narrowed exploration

In the second hunting style of the Aquila, the Aquila hovers over the target prey and then attacks. This hunting method can be expressed by the following mathematical formula:

$$X_2(t+1) = X_{best}(t) * LF(D) + X_R(t) + (y - x) * r,$$

where $LF(D)$ is the levy flight function. $X_R(t)$ is a randomly generated solution in the search space. $LF(D)$ can be calculated by using the following formula.

$$LF(D) = s * \frac{u * \sigma}{\mid v \mid^{\frac{1}{\beta}}}, \sigma = \frac{\Gamma(1+\beta) * sin(\frac{\pi\beta}{2})}{\Gamma(1+\beta) * \beta * 2^{\frac{\beta-1}{2}}}$$

where $s = 0.01$ and $\beta = 1.5$. u and v are random numbers between 0 and 1. x and y can be calculated by the following formulas:

$$x = r^* * sin(\theta), y = r^* * cos(\theta), r^* = r_1 + 0.00565 * D_1, \theta = -\omega * D_1 + \theta_1, \theta_1 = \frac{3\pi}{2},$$

where r_1 is the number of search cycles and $\omega = 0.005$. D_1 is an integer from 1 to dimension D.

4.5.3. Step 3: Expanded exploitation

In the third hunting style of the Aquila, the Aquila uses the selected area of the target to approach the prey and attack. This hunting pattern can be calculated using the following mathematical formula:

$$X_3(t+1) = (X_{best}(t) - X_M(t)) * \alpha - r + ((UB - LB) * rand + LB) * \sigma,$$

where $\alpha = 0.1$ and $\sigma = 0.1$ are the parameters of the tuning algorithm, respectively. UB and LB are the upper and lower bounds in the problem, respectively.

4.5.4. Step 4: Narrowed exploitation

In the fourth hunting method, the Aquila randomly move to attack the prey. This hunting method can be expressed by the following mathematical formula:

$$\begin{aligned}
&X_4(t+1) = QF * X_{best}(t) - (G_1 * X(t) * r) - G_2 * Levy + r * G_1,\\
&QF(t) = t^{\frac{2*rand-1}{(1-Max\_iter)^2}},\\
&G_1 = 2 * r - 1,\\
&G_2 = 2 * (1 - \frac{t}{Max\_iter}),
\end{aligned}$$

where QF is used to balance the search strategy. G_1 defines the motion parameters of the Aquila when hunting, which is a random value between -1 and 1. G_2 represents the flight slope of the Aquila when hunting.

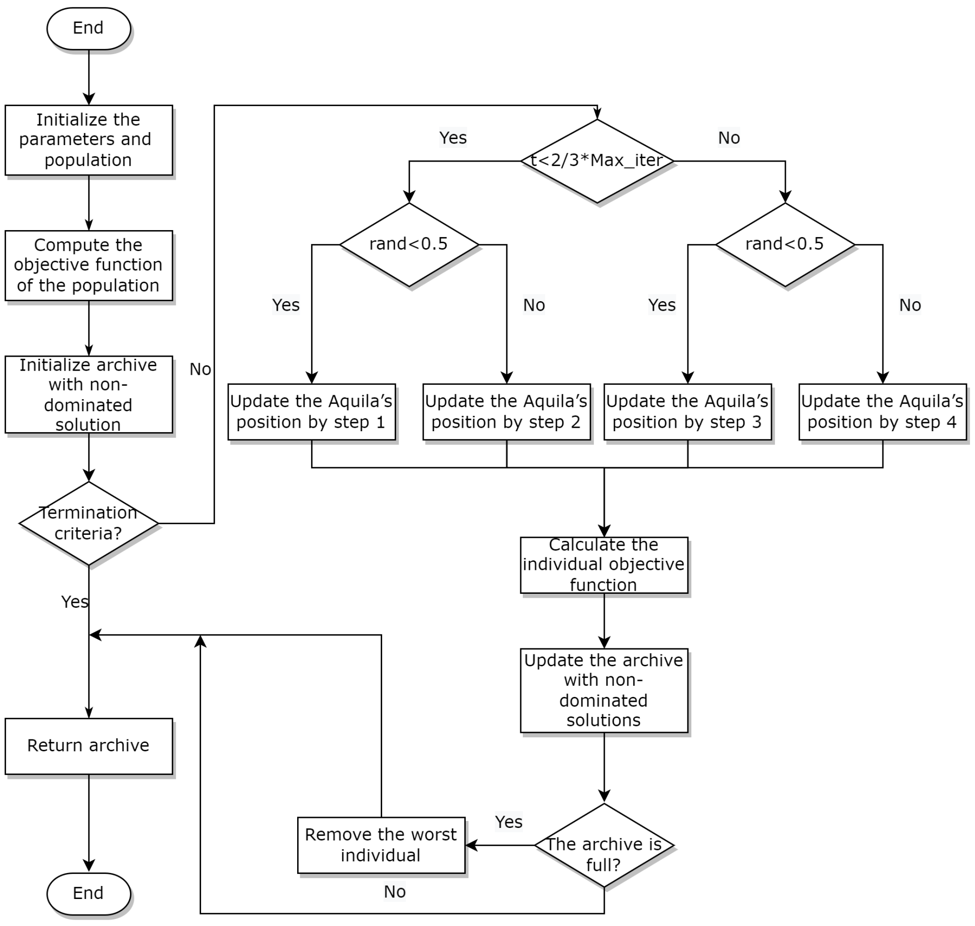

End
Initialize the parameters and population
Compute the objective function of the population
Initialize archive with non-dominated solution
Termination criteria?
No
Yes
t<2/3*Max_iter
Yes
No
rand<0.5
rand<0.5
Yes
No
Yes
No
Update the Aquila's position by step 1
Update the Aquila's position by step 2
Update the Aquila's position by step 3
Update the Aquila's position by step 4
Calculate the individual objective function
Update the archive with non-dominated solutions
The archive is full?
Yes
No
Remove the worst individual
Return archive
End

Figure 8. Multi-objective AO.

4.5.5. Multi-objective Aquila optimizer

To improve the local and global search capabilities of the AO in multi-objective optimization, we have established an archive of Pareto optimal results [32]. When the archive is full, we remove the worst individuals from the archive. The flow chart of the multi-objective AO is shown in Figure 8.

4.6. Hybrid genetic algorithm and variable neighborhood search algorithm

Here we will use the local search metaheuristic VNS to improve the NSGA-II. In the previous subsections, we defined the two algorithms separately. The NSGA-II is a classic multi-objective optimization algorithm proposed by Deb et al. in 2002 [33], and its performance has been proved in many fields. In order to enhance the local search ability of the NSGA-II, we use the VNS algorithm to further improve the solution generated by the NSGA-II. In this approach, our problem is divided into two stages; in the first stage, the GA is applied, followed by the VNS algorithm to optimize the current solution. Although the VNS algorithm can efficiently search the entire space of problems, it can deeply optimize large-scale optimization problems. However, its search process is time-consuming. Therefore, in order to balance the search time and the quality of the solution, we do not use VNS in every iteration, but use the VNS algorithm at intervals. The pseudocode of the hybrid GA and VNS algorithm is shown in Algorithm 1, where N is the population size.

Algorithm 1 Hybrid GA and VNS

```
1: Parameters to initialize GA and VNS include $MaxIter$ (maximum number of iterations), $N$ (population size), $P_c$ (crossover probability) and $P_m$ (mutation probability);
2: Initialize the individual and compute the objective function of the individual;
3: t = 0;
4: while $t \leq MaxIter$ do
5:     Run the NSGA-II;
6:     if $t \bmod D = 0$ then
7:         Run the VNS algorithm;;
8:     end if
9: end while
10: Returns the non-dominant solutions.
```

4.7. Hybrid genetic algorithm and Aquila optimizer

In this section, we use the traditional algorithm, i.e., a GA, to improve the recently proposed optimization algorithm, the AO, as a new metaheuristic hybrid algorithm. In the front, we defined the multi-objective optimization algorithms corresponding to these two algorithms respectively. Next, we propose a hybrid algorithm consisting of a GA and the AO. Briefly, we divide the population size N into two groups (N_1 and N_2), where one group is optimized with the GA and one group is optimized with the AO algorithm. In our hybrid algorithm, we first use the AO algorithm to optimize some individuals, and then we randomly select two individuals from all of the AO individuals and use the crossover and mutation algorithm to generate the remaining individuals. To describe our algorithm in more detail, the definition of the pseudocode is shown in Algorithm 2.

Algorithm 2 Hybrid GA and AO

```
1:  Parameters to initialize GA and AO include $MaxIter$ (maximum number of iterations), $N$ (population size), $P_c$ (crossover probability) and $P_m$ (mutation probability);
2:  Initialize the individual $X$ and compute the objective function of the individual;
3:  $t = 0$;
4:  Update the Archive with non-dominant solutions;
5:  while $t \leq MaxIter$ do
6:      Randomly divide the population $X$ into two groups $X_1$ and $X_2$;
7:      for each individual in $X_1$ do
8:          Update $X_1$ by running AO algorithm;
9:      end for
10:     for each individual in $X_2$ do
11:         Randomly select two individuals from $X$;
12:         Update $X_2$ by crossover and mutation operators;
13:     end for
14:     Merge $X_1$ and $X_2$;
15:     Generate a new population with population size $N$ according to the crowding distance;
16:     Update the Archive with non-dominant solutions;
17:     t = t + 1;
18: end while
19: Returns the Archive
```

5. Experimental results and analysis

In this section, we describe the use of a series of data to test our proposed hybrid algorithm. In order to better test the performance of the algorithm, we used the Taguchi method to adjust the parameters of all of the algorithms involved in the test. At the same time, in order to evaluate the performance of the algorithm, we introduce some evaluation indicators of the multi-objective optimization algorithm. We compare the two proposed hybrid algorithms and the basic algorithms that make up these two hybrid algorithms, including the AO, GA and VNS. Going a step further, we compare the algorithm with an exact algorithm for multi-objective optimization in some instances. We used PyCharm 2019.3.2 software to call CPLEX to solve this exact algorithm. The experimental environment of this study was as follows: 64-bit Windows 10; 2.80 GHz Intel i7-1165 CPU; 16G memory; programming environment: PyCharm 2019.3.2 x64.

5.1. Test dataset

In our simulation experiments, we used the basic data from the 2021 Huawei Cup Graduate Mathematical Modeling Question F. The dataset consists of Data A and Data B, where Data A is a small scale dataset and Data B is a large scale dataset. We only used the large-scale dataset, Data B, in our simulation experiments. In dataset Data B, there are a total of 13,954 flights and 465 pilots in a one-month period. In the pilot's information table, EmpNo., Captain qualification, FirstOfficer qualification, Deadhead, Base, DutyCostPerHr and ParingCostPerHr are included. At the same time, we randomly assigned qualification information and language information to each flight and pilot. We also randomly generated for each pilot his satisfaction with flight pairings. We grouped these flights into a pairing of 1784 flights. We divided these flight pairings and pilots into 10 test instances, as shown in Table 6. In Table 6, we call ZT1 to ZT5 small scale instances and ZT6 to ZT10 large scale instances.

Table 6. Test problems.

Instance	No. pairings	No. pilots	**Instance**	No. pairings	No. pilots
ZT1	60	97	ZT6	594	386
ZT2	75	192	ZT7	446	348
ZT3	148	289	ZT8	669	465
ZT4	198	308	ZT9	595	371
ZT5	222	330	ZT10	357	232

5.2. *Taguchi method*

In order to make several optimization algorithms reach the best state, we will adjust the parameters of the algorithm. In this subsection, we will list the experimental design for tuning the parameters of the algorithm. We will use Taguchi's method [34] for tuning the experimental parameters of the algorithm. So far, this approach has played an important role in parameter tuning in several fields [35,36].

Table 7. Impact factors and their levels.

Algorithm	Factor	Level
GA	Maximum Iteration (Max_iter)	A = 100, B = 150, C = 200, D = 250, E = 300
	Number of populations (N)	A = 100, B = 125, C = 150, D = 175, E = 200
	Crossover probability (P_c)	A = 0.7, B = 0.75, C = 0.8, D = 0.85, E = 0.9
	Mutation probability (P_m)	A = 0.1, B = 0.15, C = 0.2, D = 0.25, E = 0.3
AO	Maximum Iteration (Max_iter)	A = 100, B = 150, C = 200, D = 250, E = 300
	Number of populations (N)	A = 100, B = 125, C = 150, D = 175, E = 200
VNS	Maximum Iteration (Max_iter)	A = 100, B = 150, C = 200, D = 250, E = 300
	Number of populations (N)	A = 100, B = 125, C = 150, D = 175, E = 200
GA-VNS	Maximum Iteration (Max_iter)	A = 100, B = 150, C = 200, D = 250, E = 300
	Number of populations (N)	A = 100, B = 125, C = 150, D = 175, E = 200
	Crossover probability (P_c)	A = 0.7, B = 0.75, C = 0.8, D = 0.85, E = 0.9
	Mutation probability (P_m)	A = 0.1, B = 0.15, C = 0.2, D = 0.25, E = 0.3
	Interval iterations (D)	A = 10, B = 20, C = 30, D = 40, E = 50
AOGA	Maximum Iteration (Max_iter)	A = 100, B = 150, C = 200, D = 250, E = 300
	Number of populations (N)	A = 100, B = 125, C = 150, D = 175, E = 200
	Crossover probability (P_c)	A = 0.7, B = 0.75, C = 0.8, D = 0.85, E = 0.9
	Mutation probability (P_m)	A = 0.1, B = 0.15, C = 0.2, D = 0.25, E = 0.3

Table 8. Orthogonal array ZT1 for AO and VNS.

ZT1	Max_iter	N
1	1	1
2	1	2
3	1	3
4	1	4
5	1	5
6	2	1
7	2	2
8	2	3
9	2	4
10	2	5
11	3	1
12	3	2
13	3	3
14	3	4
15	3	5
16	4	1
17	4	2
18	4	3
19	4	4
20	4	5
21	5	1
22	5	2
23	5	3
24	5	4
25	5	5

With the Taguchi method, we used the signal-to-noise (S/N) ratio for experimental analysis. The S/N ratio is the standard for evaluating the stability of the system, that is to say, the larger the ratio, the smaller the impact of noise on the system. The S/N ratio measures quality characteristics that deviate from expected values. It can be defined as follows:

$$S/N = -10log(MSD),$$

where MSD is the mean squared deviation of the mass characteristic.

In Table 7, we provide the impact factors for five algorithms, while providing five levels for each factor. We should note that the Taguchi method uses a set of Taguchi orthogonal arrays to control the running time of the algorithm, and that the Taguchi method reduces the total number of tests compared to the full-experiment factorial method. The experimental combinations we designed for the five algorithms are shown in Tables 8–10. In other words, each algorithm requires 25 sets of experiments. To save algorithm time, we conducte all tests on a small-scale instance ZT1. In order to get the best level of each algorithm, we used Minitab software to analyze the S/N ratio. The optimal parameter values using the selected algorithm are reported in Table 11.

5.3. Assessment metrics

In this subsection, we describe the use of metrics to evaluate the quality of the obtained non-dominant solutions of the five metaheuristics. The four evaluation indicators are the percentage of domination (POD), number of Pareto solutions (NPS), data envelopment analysis (DEA) and diversification metric (DM). Below we briefly describe these four evaluation indicators.

Table 9. Orthogonal array ZT1 for the AOGA and GA.

ZT1	Max_iter	N	P_c	P_m
1	1	1	1	1
2	1	2	2	2
3	1	3	3	3
4	1	4	4	4
5	1	5	5	5
6	2	1	2	3
7	2	2	3	4
8	2	3	4	5
9	2	4	5	1
10	2	5	1	2
11	3	1	3	5
12	3	2	4	1
13	3	3	5	2
14	3	4	1	3
15	3	5	2	4
16	4	1	4	2
17	4	2	5	3
18	4	3	1	4
19	4	4	2	5
20	4	5	3	1
21	5	1	5	4
22	5	2	1	5
23	5	3	2	1
24	5	4	3	2
25	5	5	4	3

POD: This metric can be used to assess the ability to dominate other algorithms [37]. POD metrics are defined in terms of coverage metrics (CM). CM measures the coverage between two algorithms, which can be calculated by using the following formula:

$$CM(X_1, X_2) = \frac{| \{x_2 \in X_2\} \mid \exists x_1 \in X_1 : x_1 \leq x_2 |}{| X_2 |}. \tag{5.1}$$

When the number of algorithms is more than two, the CM indicator is no longer applicable. At this point, all algorithms should be compared. The POD indicator can be defined by the following formula:

$$POD(X_1, X_2, ..., X_n) = \frac{|\{x_{ki} \in X_i\} \mid \exists x_{kj} \in X_1, i \in j : x_{ki} \leq x_{kj}|}{|X_j|}. \tag{5.2}$$

Table 10. Orthogonal array ZT1 for GA-VNS.

ZT1	Max_iter	N	P_c	P_m	D
1	1	1	1	1	1
2	1	2	2	2	2
3	1	3	3	3	3
4	1	4	4	4	4
5	1	5	5	5	5
6	2	1	2	3	4
7	2	2	3	4	5
8	2	3	4	5	1
9	2	4	5	1	2
10	2	5	1	2	3
11	3	1	3	5	2
12	3	2	4	1	3
13	3	3	5	2	4
14	3	4	1	3	5
15	3	5	2	4	1
16	4	1	4	2	5
17	4	2	5	3	1
18	4	3	1	4	2
19	4	4	2	5	3
20	4	5	3	1	4
21	5	1	5	4	3
22	5	2	1	5	4
23	5	3	2	1	5
24	5	4	3	2	1
25	5	5	4	3	2

Table 11. Optimal parameter values for the algorithm.

Algorithm	Parameters
GA	$max\_iter = 250$, $N = 150$, $P_c = 0.75$, $P_m = 0.15$
AO	$max\_iter = 200$, $N = 125$
VNS	$max\_iter = 150$, $N = 125$
GA-VNS	$max\_iter = 250$, $N = 125$, $P_c = 0.8$, $P_m = 0.1$, $D = 40$
AOGA	$max\_iter = 200$, $N = 125$, $P_c = 0.7$, $P_m = 0.25$

NPS: NPS computes all non-dominant solutions of the algorithm [38].

DEA: The DEA indicator can be used to determine the efficiency of the solution. For a detailed description of DEA, please refer to Reference [39].

DM: The DM measures the spread of non-dominant solutions. For the bi-objective model proposed in this paper, we can use the following formula to calculate:

$$DM = \sqrt{(f_1^{max} - f_1^{min})^2 + (f_2^{max} - f_2^{min})^2}.$$

Regarding the several assessment metrics mentioned above, the more the better.

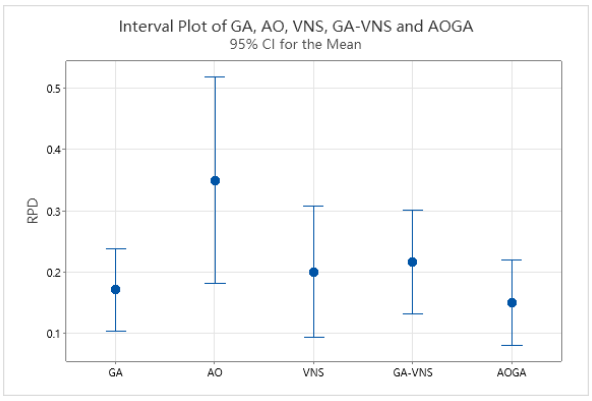
5.4. Comparison with other metaheuristic algorithms

Here, we will compare the two hybrid algorithms and the algorithms that constitute them. We use the four evaluation indicators and computational time given in the previous section to compare the algorithms.

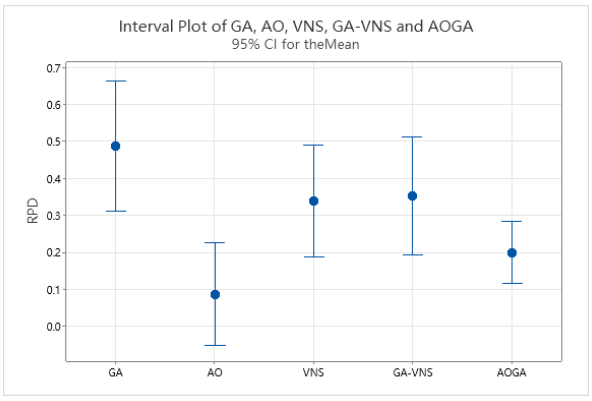
Table 12. Performance measured by the method $C(E, F)$ on instance ZT1.

Instance		ZT1	ZT2	ZT3	ZT4	ZT5	ZT6	ZT7	ZT8	ZT9	ZT10
POD	GA	0.18	0.18	0.17	0.19	0.24	0.22	0.19	0.21	0.22	0.19
	AO	0.06	0.11	0.08	0.05	0.12	0.12	0.06	0.07	0.05	0.06
	VNS	0.11	0.11	0.09	0.14	0.15	0.14	0.13	0.10	0.16	0.15
	GA-VNS	0.18	**0.22**	**0.25**	**0.23**	**0.31**	**0.27**	0.22	**0.26**	**0.27**	**0.22**
	AOGA	**0.25**	0.21	0.23	0.18	0.22	0.19	**0.26**	0.24	0.22	0.21
NPS	GA	9	12	13	13	17	19	6	26	33	21
	AO	**125**	**125**	**125**	**125**	**55**	**125**	**86**	**125**	**125**	**125**
	VNS	7	5	5	9	6	11	13	11	10	9
	GA-VNS	7	10	11	13	15	11	18	19	13	6
	AOGA	16	17	18	17	14	22	14	20	19	19
DEA	GA	0.16	0.11	0.17	0.08	0.21	0.18	0.15	0.11	0.15	0.18
	AO	0.15	0.14	0.16	0.21	0.16	0.18	0.19	0.18	0.15	0.17
	VNS	0.09	0.13	0.11	0.15	0.07	0.13	0.14	0.21	0.16	0.12
	GA-VNS	0.23	0.22	**0.23**	0.22	0.16	0.24	0.22	**0.24**	0.22	0.27
	AOGA	**0.25**	**0.24**	0.22	**0.27**	**0.29**	**0.26**	**0.25**	**0.24**	**0.27**	**0.31**
DM	GA	14,608	31,348	36,335	47,104	**78,965**	64,783	33,116	83,528	66,627	77,310
	AO	13,560	5103	12,816	6170	9480	34,586	486	5070	29,046	39,170
	VNS	14,312	11,672	31,254	27,121	15,216	51,621	32,267	34,215	42,182	51,261
	GA-VNS	11,316	**17,366**	53,710	46,900	33,040	50,840	**66,876**	91,050	**73,033**	**83,320**
	AOGA	**26,640**	14,693	**56,913**	**49,376**	33,716	**75,370**	46,010	**95,006**	49,626	59,920

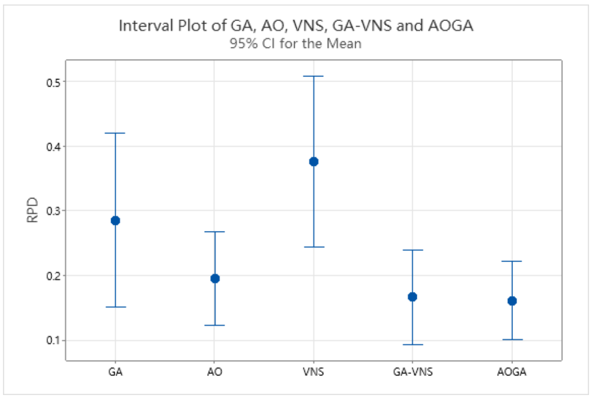
The comparison results of the algorithm on the four evaluation indicators are shown in Table 12. The search process of metaheuristic algorithm is not deterministic, so the value generated in each run may be different. Therefore, we consider the average value of each algorithm running 10 times on the same instance to be reliable. Table 12 describes the results of several metaheuristic algorithms on four evaluation indicators. The best value in each instance is shown in bold. It can be seen in Table 12 that the AO algorithm performed well on the NPS index, which means that the non-dominant solution generated by the AO algorithm is more than other algorithms. But this does not mean that the AO algorithm is superior to other algorithms as a whole, because it can be seen from the POD index that the non-dominant solution generated by the AO algorithm will be dominated by the non-dominant solution generated by other algorithms. Obviously, on the POD index, the solution generated by the GA-VNS algorithm has a strong ability to dominate other algorithms on most instances. At the same time, in terms of the DEA and DM indicators, the two hybrid algorithms we propose are better than the three basic algorithms that make up the hybrid algorithm.



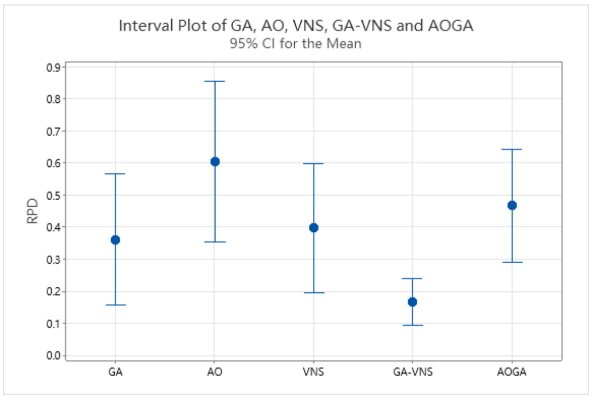
(a) POD



(b) NPS



(c) DEA



(d) DM

Figure 9. Comparison of algorithms.

In order to more accurately evaluate the multi-objective indicators on 10 instances reported by the algorithm in Table 12, we standardized the multi-objective indicators on 10 instances by using relative percentage deviation (RPD) [40]. The RPD can be calculated by the following formula:

$$RPD = \frac{|\,ALG_{sol} - BEST_{sol}\,|}{BEST_{sol}} \tag{5.3}$$

where $BEST_{sol}$ is the best solution of the algorithm in each evaluation index and ALG_{sol} is the solution of the algorithm in each evaluation index. Obviously, the smaller the RPD value, the better the performance of the algorithm. We first standardized all of the data in Table 12 based on the RPD, and then performed a statistical test on the RPD value based on the 95% confidence level. The statistical results for the four evaluation indicators on five metaheuristic algorithms are shown in Figure 9. As shown in Figure 9(a), the AOGA was the best algorithm on the indicator POD, while the AO was the worst algorithm. Based on the NPS index in Figure 9(b), the best algorithm is the AO, and the AOGA is the second best metaheuristic algorithm. However, the GA-VNS and AOGA were the best and second best metaheuristics, respectively, for the index DEA in Figure 9(c). GA-VNS is the best metaheuristic algorithm for the index DM in Figure 9(d).

Therefore, we can draw a conclusion from Figure 9 that GA-VNS is the algorithm with the best performance on both indicators, while AOGA is the algorithm with a top 2 ranking on all three indicators. In Figure 10, we show a comparison of the running times of several algorithms. It can be seen from Figure 10 that the computing time difference between GA, GA-VNS and AOGA algorithms on small-scale instances was small, but the computing time of the VNS and AO algorithms was long. However, in all instances, the VNS algorithm took the most computing time, while the GA consumed the least.

5.5. Comparison with exact methods

In this section, in order to further evaluate the reliability of the Pareto solution generated by the AOGA and GA, we also compared it with the epsilon constraint (EC) method proposed by Haimes et al. [41] Here, we regard f_1 as the main goal and f_2 as the constraint. Obviously, from the previous description of f_2, we can know that the upper bound of f_2 is 5 and the lower bound is 0. Therefore, we can convert the mathematical model as follows:

$$Min \qquad f_1 \tag{5.4}$$

subject to:

$$Eqs. \quad (3.2) \quad to \quad (3.7) \tag{5.5}$$

$$f_2 \geq \epsilon \tag{5.6}$$

$$0 \leq \epsilon \leq 5 \tag{5.7}$$

In order to compare the effectiveness of the exact algorithm and the metaheuristics algorithm we proposed, we first calculated the NPS. We will use the aforementioned CM index to evaluate the epsilon constraint method and metaheuristic algorithm. Like [40], we compared each solution of the metaheuristic algorithm with the solution of the EC and defined the modified NPS by EC (MNPSC) [40]. Here, we only ran EC method on small-scale instances ZT1 and ZT2. Table 13 reports our analysis results. We can clearly see that the AOGA had a higher proportion of effective non-dominant ratios than the GA-VNS algorithm. The effective non-dominant ratios of both algorithms exceeded 0.6, which shows that our algorithm has a good effect.

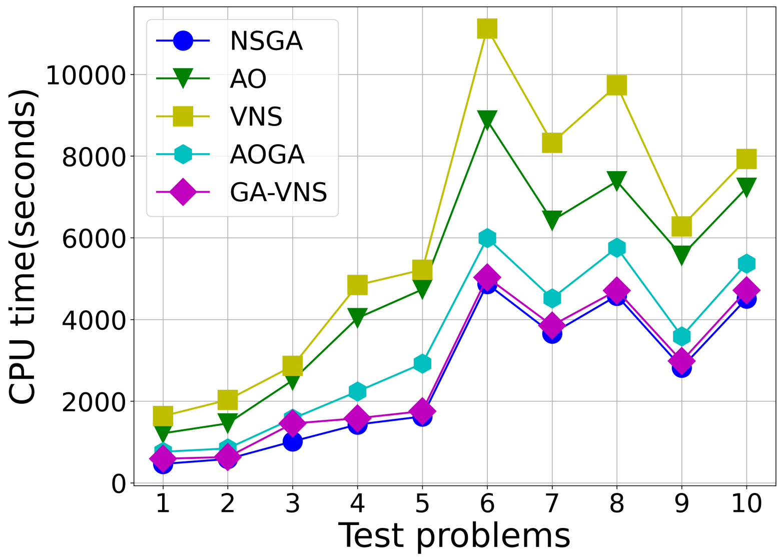


Figure 10. Comparison of calculation time of algorithms.

Table 13. Validation of the proposed algorithms.

Instance	GA-VNS			AOGA		
	NPS	MNPSEC	MNPSEC/NPS	NPS	MNPSEC	MNPSEC/NPS
ZT1	7	5	0.71	16	11	0.69
ZT2	10	6	0.60	17	13	0.77
Average	8.5	0.655	16.5	12	13	0.73

6. Conclusions

Given the importance of crew resources in airlines, we studied the problem of crew scheduling. At the same time, due to the existence of a large number of plateau airports and special airports in China, pilots with these qualifications have become very scarce. In order to be able to make full use of these resources and solve the CRP, considering fairness and satisfaction, a multi-objective framework has been defined for the CRP. This issue is complicated by the objectives of the CRP and the various constraints of aviation laws and regulations. This creates the need for efficient heuristics; therefore, we introduce two metaheuristics including the AOGA and GA-VNS in this paper. Hybrid metaheuristics were compared with other algorithms by using different criteria including CPU time and multi-objective criteria, and the analysis results report the effectiveness of the algorithm.

In conclusion, although this research contributes to the aviation industry and algorithm research, there are still some limitations. First, this study did not consider parameter uncertainty. Therefore it is a new direction to develop a stochastic optimization model in future work. In addition, another effective suggestion is to consider surrogate-assisted in the hybrid algorithm. Finally, we can apply our hybrid metaheuristic algorithm to other optimization problems, including cloud computing [42] and supply chain problems [40].

Acknowledgments

This work was supported by the Postgraduate Research and Innovation Foundation of Yunnan University (2021Z089).

Conflict of interest

All author declare no conflict of interest regarding this paper.

References

1. Civil Aviation Administration of China, *2018 Civil Aviation Industry Development Statistics Bulletin*, 2019. Available from: http://www.caac.gov.cn/XXGK/XXGK/TJSJ/201905/P0201905085195 29727887.pdf.
2. M. Ehrgott, D. M. Ryan, Constructing robust crew schedules with bicriteria optimization, *J. Multi-Criter. Decis. Anal.*, **11** (2002), 139–150. https://doi.org/10.1002/mcda.321
3. S. Zhou, Z. Zhan, Z. Chen, S. Kwong, J. Zhang, A multi-objective ant colony system algorithm for airline crew rostering problem with fairness and satisfaction, *IEEE Trans. Intell. Transp. Syst.*, **22** (2020), 6784–6798. https://doi.org/10.1109/TITS.2020.2994779
4. R. T. Marler, J. S. Arora, Survey of multi-objective optimization methods for engineering, *Struct. Multidiscip. Optim.*, **26** (2004), 369–395. https://doi.org/10.1007/s00158-003-0368-6
5. Q. Yang, L. Dan, M. Lv, J. Wu, W. Li, W. Dong, Quantitative assessment of the parameterization sensitivity of the Noah-MP land surface model with dynamic vegetation using ChinaFLUX data, *Agric. For. Meteorol.*, **307** (2021), 108542. https://doi.org/10.1016/j.agrformet.2021.108542
6. Q. Yang, L. Dan, J. Wu, R. Jiang, J. Dan, W. Li, et al., The improved freeze–thaw process of a climate-vegetation model: Calibration and validation tests in the source region of the Yellow River, *Agric. For. Meteorol.*, **123** (2018), 346–367. https://doi.org/10.1029/2017JD028050
7. J. E. Beasley, B. Cao, A tree search algorithm for the crew scheduling problem, *Eur. J. Oper. Res.*, **94** (1996), 517–526. https://doi.org/10.1016/0377-2217(95)00093-3
8. J. E. Beasley, B. Cao, A dynamic programming based algorithm for the crew scheduling problem, *Comput. Oper. Res.*, **25** (1998), 567–582. https://doi.org/10.1016/S0305-0548(98)00019-7
9. P. Lučić, D. Teodorović, Metaheuristics approach to the aircrew rostering problem, *Ann. Oper. Res.*, **155** (2007), 311–338. https://doi.org/10.1007/s10479-007-0216-y
10. B. Maenhout, M. Vanhoucke, A hybrid scatter search heuristic for personalized crew rostering in the airline industry, *Eur. J. Oper. Res.*, **206** (2010), 155–167. https://doi.org/10.1016/j.ejor.2010.01.040
11. R. Hadianti, K. Novianingsih, S. Uttunggadewa, K. Sidarto, N. Sumarti, E. Soewono, Optimization model for an airline crew rostering problem: Case of Garuda Indonesia, *J. Math. Fundam. Sci.*, **45** (2013), 218–234. https://doi.org/10.5614/j.math.fund.sci.2013.45.3.2

12. F. Quesnel, G. Desaulniers, F. Soumis, Improving air crew rostering by considering crew preferences in the crew pairing problem, *Transp. Sci.*, **54** (2020), 97–114. https://doi.org/10.1287/trsc.2019.0913

13. F. Quesnel, A. Wu, G. Desaulniers, F. Soumis, Deep-learning-based partial pricing in a branch-and-price algorithm for personalized crew rostering, *Comput. Oper. Res.*, **138** (2022), 105554. https://doi.org/10.1016/j.cor.2021.105554

14. B. Deng, An improved honey badger algorithm by genetic algorithm and levy flight distribution for solving airline crew rostering problem, *IEEE Access*, **10** (2022), 108075–108088. https://doi.org/10.1109/ACCESS.2022.3213066

15. N. Souai, J. Teghem, Genetic algorithm based approach for the integrated airline crew-pairing and rostering problem, *Eur. J. Oper. Res.*, **199** (2009), 674–683. https://doi.org/10.1016/j.ejor.2007.10.065

16. M. Saddoune, G. Desaulniers, I. Elhallaoui, F. Soumis, A. Fathollahi-Fard, Integrated airline crew pairing and crew assignment by dynamic constraint aggregation, *Transp. Sci.*, **46** (2012), 39–55. https://doi.org/10.1287/trsc.1110.0379

17. V. Zeighami, M. Saddoune, F. Soumis, Alternating Lagrangian decomposition for integrated airline crew scheduling problem, *Eur. J. Oper. Res.*, **287** (2020), 211–224. https://doi.org/10.1016/j.ejor.2020.05.005

18. A. M. Fathollahi-Fard, A. Ahmadi, F. Goodarzian, N. Cheikhrouhou, A bi-objective home healthcare routing and scheduling problem considering patients' satisfaction in a fuzzy environment, *Appl. Soft Comput.*, **93** (2020), 106385. https://doi.org/10.1016/j.asoc.2020.106385

19. Z. Sazvar, S. Mirzapour Al-E-Hashem, A. Baboli, M. A. Jokar, A bi-objective stochastic programming model for a centralized green supply chain with deteriorating products, *Int. J. Prod. Econ.*, **150** (2014), 140–154. https://doi.org/10.1016/j.ijpe.2013.12.023

20. J. Pasha, A. L. Nwodu, A. M. Fathollahi-Fard, G. Tian, Z. Li, H. Wang, et al., Exact and metaheuristic algorithms for the vehicle routing problem with a factory-in-a-box in multi-objective settings, *Adv. Eng. Inf.*, **52** (2022), 101623. https://doi.org/10.1016/j.aei.2022.101623

21. A. M. Fathollahi-Fard, L. Woodward, O. Akhrif, Sustainable distributed permutation flow-shop scheduling model based on a triple bottom line concept, *J. Ind. Inf. Integr.*, **24** (2021), 100233. https://doi.org/10.1016/j.jii.2021.100233

22. E. K. Burke, P. De Causmaecker, G. De Maere, J. Mulder, M. Paelinck, G. V. Berghe, A multi-objective approach for robust airline scheduling, *Comput. Oper. Res.*, **37** (2010), 822–832. https://doi.org/10.1016/j.cor.2009.03.026

23. P. Chutima, K. Arayikanon, Many-objective low-cost airline cockpit crew rostering optimisation, *Comput. Ind. Eng.*, **150** (2020), 106844. https://doi.org/10.1016/j.cie.2020.106844

24. V. Baradaran, A. H. Hosseinian, A multi-objective mathematical formulation for the airline crew scheduling problem: MODE and NSGA-II solution approaches, *J. Ind. Manage. Perspect.*, **11** (2021), 247–269. https://doi.org/10.52547/jimp.11.1.247

25. Q. Yang, H. Zuo, W. Li, Land surface model and particle swarm optimization algorithm based on the model-optimization method for improving soil moisture simulation in a semi-arid region, *Plos One*, **11** (2016), e0151576. https://doi.org/10.1371/journal.pone.0151576

26. Q. Yang, J. Wu, Y. Li, W. Li, L. Wang, Y. Yang, Using the particle swarm optimization algorithm to calibrate the parameters relating to the turbulent flux in the surface layer in the source region of the Yellow River, *Agric. For. Meteorol.*, **232** (2017), 606–622. https://doi.org/10.1016/j.agrformet.2016.10.019

27. L. Abualigah, D. Yousri, M. A. Al-Qaness, A. H. Gandomi, Aquila optimizer: A novel meta-heuristic optimization algorithm, *Comput. Ind. Eng.*, **157** (2021), 107250. https://doi.org/10.1016/j.cie.2021.107250

28. B. Naderi, R. Tavakkoli-Moghaddam, M. Khalili, Electromagnetism-like mechanism and simulated annealing algorithms for flowshop scheduling problems minimizing the total weighted tardiness and makespan, *Knowl. Based Syst.*, **23** (2010), 77–85. https://doi.org/10.1016/j.knosys.2009.06.002

29. B. Vahdani, M. Zandieh, Scheduling trucks in cross-docking systems: Robust meta-heuristics, *Comput. Ind. Eng.*, **58** (2010), 12–24. https://doi.org/10.1016/j.cie.2009.06.006

30. M. Abd Elaziz, A. Dahou, N. A. Alsaleh, A. H. Elsheikh, A. I. Saba, M. Ahmadein, Boosting COVID-19 image classification using MobileNetV3 and aquila optimizer algorithm, *Entropy*, **23** (2021), 1383. https://doi.org/10.3390/e23111383

31. A. M. AlRassas, M. A. Al-qaness, A. A. Ewees, S. Ren, M. Abd Elaziz, R. Damaševičius, et al., Optimized ANFIS model using Aquila Optimizer for oil production forecasting, *Processes*, **9** (2021), 1194. https://doi.org/10.3390/pr9071194

32. Q. Xing, J. Wang, H. Lu, S. Wang, Research of a novel short-term wind forecasting system based on multi-objective Aquila optimizer for point and interval forecast, *Energy Convers. Manage.*, **263** (2022), 115583. https://doi.org/10.1016/j.enconman.2022.115583

33. K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, *IEEE Trans. Evol. Comput.*, **6** (2002), 182–197. https://doi.org/10.1007/s00158-003-0368-6

34. G. Taguchi, *Introduction to Quality Engineering: Designing Quality into Products and Processes*, 1986.

35. M. Rezaei, M. Afsahi, M. Shafiee, M. Patriksson, A bi-objective optimization framework for designing an efficient fuel supply chain network in post-earthquakes, *Comput. Ind. Eng.*, **147** (2020), 106654. https://doi.org/10.1016/j.cie.2020.106654

36. N. Janatyan, M. Zandieh, A. Alem-Tabriz, M. Rabieh, A robust optimization model for sustainable pharmaceutical distribution network design: A case study, *Ann. Oper. Res.*, **46** (2021), 1–20. https://doi.org/10.1007/s10479-020-03900-5

37. K. Govindan, A. Jafarian, M. E. Azbari, T. Choi, Optimal bi-objective redundancy allocation for systems reliability and risk management, *IEEE Trans. Cybern.*, **46** (2015), 1735–1748. https://doi.org/10.1109/TCYB.2014.2382666

38. F. Goodarzian, A. A. Taleizadeh, P. Ghasemi, A. Abraham, An integrated sustainable medical supply chain network during COVID-19, *Eng. Appl. Artif. Intell.*, **100** (2021), 104188. https://doi.org/10.1016/j.engappai.2021.104188

39. G. R. Amin, M. Toloo, Finding the most efficient DMUs in DEA: An improved integrated model, *Comput. Ind. Eng.*, **52** (2007), 71–77. https://doi.org/10.1016/j.cie.2006.10.003

40. P. Seydanlou, F. Jolai, R. Tavakkoli-Moghaddam, A. Fathollahi-Fard, A multi-objective optimization framework for a sustainable closed-loop supply chain network in the olive industry: Hybrid meta-heuristic algorithms, *Expert Syst. Appl.*, **203** (2022), 117566. https://doi.org/10.1016/j.eswa.2022.117566

41. Y. Haimes, On a bicriterion formulation of the problems of integrated system identification and system optimization, *IEEE Trans. Syst. Man Cybern.*, **1** (1971), 296–297.

42. X. Liu, X. Zhang, W. Li, X. Zhang, Swarm optimization algorithms applied to multi-resource fair allocation in heterogeneous cloud computing systems, *Computing*, **99** (2017), 1231–1255. https://doi.org/10.1007/s00607-017-0561-x