Hindawi
Journal of Mathematics
Volume 2022, Article ID 8806135, 13 pages
https://doi.org/10.1155/2022/8806135

Research Article

A Game-Theoretic Approach for the Robust Daily Aircraft Routing Problem

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

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

Correspondence should be addressed to Bin Deng; dengbin96@126.com

Received 25 August 2022; Revised 15 October 2022; Accepted 14 November 2022; Published 6 December 2022

Academic Editor: Efthymios G. Tsionas

In the operation of airlines, the most important link is determining the route and scheduling of aircraft. The key to this problem is to input the flight segment and aircraft type and finally determine all flight segments for each aircraft. In this paper, we focus on finding feasible, robust scheduling for various uncertainties in the flight process. This paper presents a new robust integer mathematical model based on game theory that considers daily aircraft routing. Then, in order to find the suboptimal solution to a large-scale integer programming problem in a limited amount of time, a heuristic algorithm integrating a column generation algorithm and variable domain search is introduced. In addition, we use the data of a Chinese airline to verify, and the experimental results show that the model proposed by us is more robust than the model in general.

1. Introduction

With the increasingly fierce competition in the aviation industry, an effective decision is very important to the profitability of airlines. However, designing an aviation network is a very complex task. Therefore, many researchers generally divide the design of the entire aviation network into a series of subproblems, including route traffic volume prediction, flight planning, aircraft assignment, crew pairing, and crew rostering. The route traffic volume forecast is based on the traffic volume of the national air transport market, the route market, and the economic situation of a region. Flight planning is to generate flight schedules based on the previously predicted route traffic volume. Aircraft scheduling is to generate an aircraft scheduling plan based on the aircraft type selected for the flight plan and other relevant constraints. Duty scheduling is to generate a flight segment pairing that meets the requirements of a flight crew's one-day mission according to the required flight time and laws and regulations in the flight plan, that is, the duty schedule. The crew rostering generates the flight schedule in accordance with laws and regulations based on the generated flight pairings and crew members. In this paper, we focus on the aircraft assignment phase.

Although there are more and more routes and the route structure is more and more complex due to the restrictions on airspace resources, airport facilities, and the various conditions of airlines, coupled with weather reasons, flight delays and even flight cancellations are becoming more and more frequent, which has gradually become a major problem in the development of civil aviation. According to the data of the Civil Aviation Administration of China, with the increase in the number of flights and routes, the normal rate of flights has shown a downward trend. Flight plans with strong anti-interference can not only reduce flight delay to a certain extent but also save some costs for airlines to a certain extent. Whether the flight plan is antijamming is mainly reflected in its robustness. If a flight is delayed for some reason, it is likely to cause successive delays of subsequent flights. Therefore, it is of great practical significance and value to develop a reasonable aircraft scheduling plan based on the actual operation to reduce the airline delay rate. At present, civil aviation usually divides flight plans into two categories according to their functions during operation. On the one hand, it is an "advance strategy," that is, for the prepared flight plan, the transit time should be adjusted appropriately according to the actual operation to absorb the departure delay caused by previous flights as much as possible so as to improve the robustness of flight plans. On the other hand, it is an "after the fact strategy," which is to take certain measures and schemes in time after flight delays, such as aircraft transfer, aircraft exchange, or even cancellation of flights, so as to avoid causing larger delays. In this paper, we mainly solve a daily aircraft route problem (DARP), including route planning and route assignment. The advanced strategy is mainly used to adjust flight cost and flight plan robustness and try to improve the robustness of the flight plan on the basis of cost control.

1.1. Literature Review. In fact, the daily flight route problem is a classical routing problem. Therefore, in this section, we first review the literature on daily aircraft route problems, then review the application of game theory, and finally review the application of meta-heuristic algorithms to engineering problems.

This problem was first proposed in 1971 by Levin [1], who modeled the problem as a 0–1 programming model and solved large-scale instances using Land-and-Doig techniques. Desaulniers et al. [2] model the problem as a set of daily aircraft route allocation models and multicommodity flow models to improve airline efficiency and reduce costs, and solve both models using column generation and Dantzig–Wolfe decomposition, respectively. To solve this model, an iterative solution method was proposed. In order to solve the fleet allocation problem and the aircraft routing problem at the same time, Barnhart et al. [3] proposed a model and solution method to solve the two problems simultaneously. Sosnowska [4] successfully solved the flight data of a medium-sized company using a method based on a simulated annealing algorithm and GRASP. Mercier and Soumis [5] synthesized the aircraft routing problem and the crew scheduling problem, proposed a basic ensemble model, and solved this large-scale optimization model based on the Benders decomposition. Weide et al. [6] integrated the aircraft routing problem and the crew scheduling problem and established a large-scale 0–1 programming model considering the robustness of the problem. Jamili [7] proposed a mixed-integer programming model that integrates aircraft robustness and a hybrid heuristic that yields a more efficient solution. Kenan et al. [8] propose a two-stage planning model based on the uncertainty of demand and fare and solve the model using the method of sample average approximation. Considering the uncertainty of the data, Cadarso and de Celis [9] propose a large-scale mathematical model, which is solved by the Benders decomposition approach. Cui et al. [10] proposed 3 models, improved the VNS algorithm, compared the experimental results with commercial solvers, and verified the effectiveness of the algorithm. Si et al. [11] proposed a multicommodity flow model and an arc flow model, improved the column generation algorithm, and significantly reduced the running time. Xu et al. [12] proposed a mixed integer programming model considering the influence of propagation delay and flight retiming decisions and proposed column generation as well as variable neighborhood search for a solution. Şafak et al. [13] proposed a new two-stage stochastic decision-dependency programming model for airline network expansion. In fact, the daily flight route problem for aircraft is a routing problem. For routing problems, many studies were conducted during the new epidemic, especially on the supply chain problem [14].

Based on the previous work, the main purpose of this paper is to find a schedule that ensures the lowest cost incurred when some flights are delayed. This paper mainly develops a new method to deal with flight delays caused by various uncertain factors to maintain the stability of the system. Based on the abovementioned remarks, we establish a robust optimization model based on game theory.

Game theory is a commonly used optimization method. Game theory has been used in many fields, such as transportation and power systems. Lima et al. [15] constructed a cooperative game theory model based on the cooperative game theory framework, found a loss allocation solution, and, compared with other traditional loss allocation methods in transmission power systems. In [16], a benefit distribution method is proposed using a cooperative game, and the result shows that power producers will get more profits by cooperating in competition. In [17], to ensure sufficient power generation, a game theory-based approach to power system reserves is proposed, with planners and nature as two players. Compared with traditional methods, the proposed game theory method has better robustness and higher efficiency. In [18, 19], based on a game theory framework, a robust railway transportation network is designed for line failures.

At the same time, because it is very difficult to solve the daily flight route problem with an accurate algorithm, this paper combines the column generation algorithm and a meta-heuristic algorithm to solve it. The meta-heuristic algorithm can obtain a feasible solution close to the optimal solution in a limited time and has been widely used in engineering problems. For example, Alinaghian and Goli [20] used an improved harmonious search algorithm to solve the configuration problem of temporary medical centers in rural areas in crisis. Yang et al. [21] calibrated surface turbulence-related parameters in the source area of the Yellow River using a particle swarm optimization algorithm. Yang et al. [22] used a particle swarm optimization algorithm to effectively calibrate freezing and thawing-related parameters and improve simulation accuracy.

Based on the abovementioned literature review, according to the author's current knowledge, although some scholars have studied integrated aircraft routing and scheduling before, they have not yet used the game theory method for research. As can be seen from the previous literature on game theory, game theory is an effective way to deal with uncertain problems. At the same time, a meta-heuristic algorithm is an effective method to solve these complex models.

1.2. Contribution of the Paper. This paper presents a new model of integrated aircraft routing and scheduling that considers fleet assignment. In addition, there are many uncertain factors that cause flight delays, including weather and air traffic control. Therefore, we introduce a robust method based on game theory into the model. Finally, because the model is difficult to find the optimal solution in a limited time, a method integrating column generation and variable neighborhood search is proposed to solve this model.

1.3. Outline. The main parts of this paper are as follows: In Section 2, the problem is described and a new integer programming model is established. In Section 3, we propose a method to solve large-scale integer programming by integrating column generation and variable neighborhood search. Sections 4 and 5 are numerical experiments and conclusions, respectively.

2. Problem Definition

DARP requires aircraft to be allocated to each flight segment to minimize cost. The takeoff time and landing time of each flight segment have been fixed. If two flight segments can be connected, the arrival point of one flight segment is equal to the departure point of the other flight segment, and the minimum transfer time requirement is met. In the problem of the daily flight route of the aircraft, the delay of the flight is usually related to two factors: one is due to weather reasons, traffic reasons, passenger reasons, etc.; we call it a flight delay caused by uncertainty, and the second is the next flight delay caused by the delay of the previous flight; we call it delays due to the spread of the flight. In this section, we will robustly optimize DARP based on these two aspects. We model the DARP problem. This problem will result in a schedule that optimizes some utility functions when there are no flight delays. In this paper, we assume a robust schedule that minimizes the cost incurred by the delay in the case of flight delays.

Based on the abovementioned remarks, we now formulate a DARP problem model. Let $F(g)$ be the utility function of a DARP problem for any schedule $g \in G$, where G is the set of all feasible schedules. The construction of the network is described in Section 2.1. In the following, we use the symbol v to refer to the nodes in the network graph. We define V to be the set of all nodes in the network graph, and we assume that all flight segments are delayable. The utility function of the network is affected by flight delays. Let $F(g, v)$ be the utility function when segment $v \in V$ is delayed. Our usual approach to dealing with this kind of uncertainty is to find a network with the least cost of delay in the worst case. At this point, our problem can become $\min_{g\in G}\max_{v\in V} F(g, v)$.

Definition 1. Let G be the set of all feasible schedules let V be the set of all flight segments, and let $f(g, v)$ be the utility function where flight $v \in V$ is delayed. If there exists $g^* \in G$ such that $\min_{g\in G}\max_{v\in V} F(g, v) = F(g^*, v)$, we consider $g^* \in G$ to be an optimal robust schedule.

2.1. Deterministic Models for DARP

2.1.1. Network Construction. We define V to be the set of all legs. Each leg $v \in V$ has a departure time d_v and an arrival time a_v. The minimum connection time between two legs is t. For any two legs $v_1, v_2 \in V$, if $a_{v_2} - d_{v_1} > t$ and v_2 's arrival airport is v_2's departure airport; then, v_1, v_2 can connect, that is, there is an arc e_{v_1,v_2}, Define E as the set of all arcs, then a directed graph $G(E, V)$ can be constructed.

We first build a deterministic model that minimizes the total cost when all flights are not delayed.

2.1.2. Sets, Parameters, and Variables

A: A set of all matching routes in the plane and flight connection network.

V: A set of all legs that need to be scheduled.

K: A set of all available aircraft.

x_a: Binary variable. 1 if matching $a \in A$ is selected; 0 otherwise.

c_a: The cost of matching $a \in A$.

α_{av}: Binary variable. 1 if match $a \in A$ contains the leg v; 0 otherwise.

β_{ak}: Binary variable. 1 if the match $a \in A$ contains aircraft k; 0 otherwise.

We can get the following integer linear programming (model 1):

$$\text{Min} \sum_{a\in A} c_a x_a, \tag{1}$$

subject to

$$\sum_{a\in A} \alpha_{av} x_a = 1, \quad \forall v \in V, \tag{2}$$

$$\sum_{a\in A} \beta_{ak} x_a \leq 1, \quad \forall k \in K, \tag{3}$$

$$x_a \in \{0, 1\}, \quad \forall a \in A. \tag{4}$$

In the abovementioned linear programming, the objective function (1) is to minimize the total cost when all flights are not delayed. Constraint (2) states that each flight segment can only be assigned to one aircraft. The constraint (3) specifies that each aircraft can perform at most one route per day. The decision variable x_a is equal to 1 if the matching variable A is in the solution, 0 otherwise.

2.2. Problem Expansion. In this subsection, a robust optimization model based on game theory is proposed to deal with the uncertainty of flight delays. Under the framework of the proposed model, two players are virtualized: one is the system maker (hereinafter referred to as player 1), and one is the attacker (hereafter also referred to as player 2). Flight delays are caused by influences such as weather. The scheduling problem of daily aircraft routes is formulated.

2.2.1. A max-min Game Model under Uncertainty. Different from other robust optimization models of aircraft routes, an uncertain robust optimization model is established based on game theory. Taking player 1 and player 2 as the two participants, the constructed min-max game model is as follows:

$$\left\{\min_g F(g,v), \max_v F(g,v)\right\}, \tag{5}$$

subject to

$$\begin{gathered} G(x,g,v) \leq 0, \\ g \in G, \\ v \in V, \end{gathered} \tag{6}$$

where x is the decision variable, g is the decision of player 1, v is the decision of player 2, $F(g,v)$ is the payoff function, and $G(x,g,v)$ is the constraint, G and V are the strategy sets of player 1 and player 2, respectively. In the abovementioned model, player 1 wants to minimize function F by changing g, and player 2 wants to maximize function F by changing v. From a game theory perspective, we can describe it as a noncooperative two-player zero-sum game problem. It is known from knowledge of game theory that not all two-player zero-sum games have a pure strategy. Nash equilibrium (i.e., $\min_g \max_v F(g,v) = \max_v \min_g F(g,v)$. Therefore, we adopt the min-max game model widely used in practice, as follows:

$$\min_g \max_v F(g,v), \tag{7}$$

subject to

$$G(x,g,v) \leq 0. \tag{8}$$

In the abovementioned equation, there is always a solution (g,v). In engineering technology, there are the following meanings:

(a) Player 1's best strategy can handle when player 2 chooses the strategy that makes it the worst.

(b) In our problem, our decision order is that player 1 chooses the strategy first and player 2 chooses the strategy second. Player 1 needs to presuppose that player 2 chooses the worst strategy when formulating a strategy, so it is reasonable to formulate the decision problem as a min-max model. Since this is player 2's uncertainty, player 1's best strategy is to choose the worst strategy for player 2.

2.2.2. Modeling the Daily Airplane Route Problem as a Min-Max Game. The purpose of robust optimization of the daily aircraft route problem is to find a schedule that minimizes the cost of the system and keeps the stability of the system within a certain range. In the game model, we can think of player 2 as an attacker who wants to make the schedule made by player 1 less stable due to the uncertainty of flight delays. In this model, one player's gain causes another player's loss, so we can simply model the model as a two-player zero-sum game. In the game model, it usually includes three elements: player set, strategy set, and payoff function. These three elements are described in the following detail:

(1) Player set: In the daily plane route problem, two players are involved, the aforementioned player 1 and player 2. Player 1 is the real maker of the schedule, whose main purpose is to find the schedule with the least cost. Player 2 is a virtual player with strong uncertainty, mainly affected by factors such as weather.

(2) Strategy set: Player 1's policy set is all feasible scheduling networks, i.e., $g \in G$, player 2's strategy set is all flight segments, i.e., $v \in V$.

(3) Payoff function: Player 1's goal is to find the schedule with the lowest cost while keeping the reliability of the system within a certain range. Therefore, the function value of the scheduling network is taken as the cost of player 1, and its strategy is to minimize the cost. Player 2 is deteriorating the system, and its strategy is to maximize cost.

In the daily aircraft route problem, when player 2 chooses to attack a certain flight v, the subsequent flight may also be delayed due to the spread, and this delay probability can be obtained through experience. We adopt the approach proposed by Cui et al. [10], that the probability of flight delay is related to the transit time between two flights. We assume that t_1 is the minimum connecting time specified by the Civil Aviation Administration and m_i is the probability that the connection time is between t_i and t_{i+1}. When the connection time is greater than t_{k+1}, the probability of delay is m_{k+1}, So the probability of delay is

$$p = \begin{cases} \prod_{r=1}^{n} q_r, & n \geq 1, \\ 0, & \text{otherwise}, \end{cases} \tag{9}$$

where n is the number of flights after flight i in the aircraft's route, and $q_r = m_i$ refers to the probability of the rth delay time after flight i.

Example 1. Assuming that there is an aircraft route such as $[F1, F2, F3, F4]$, when $F1$ is attacked and delayed, that is, segment $F1$ has been determined to be delayed. At this time, it is assumed that the delay probabilities caused by the previous flight are p_{F2}, p_{F3}, p_{F4}, Then, the expected total additional cost of this flight route due to delay is $c = c_{F1} + c_{F2}p_{F2} + c_{F3}p_{F2}p_{F3} + c_{F4}p_{F2}p_{F3}p_{F4}$.

Sets, parameters, and variables are as follows:

c_v: Additional costs incurred by flight $v \in V$ being delayed.

p_v: The probability that a leg $v \in V$ is delayed.

So, under uncertainty, we have the following linear programming (model 2):

$$\min_g \max_v \sum_{a \in A} c_a x_a + \sum_{v \in V} p_v c_v, \tag{10}$$

subject to

$$\begin{aligned}&\sum_{a\in A}\alpha_{av}x_a = 1, \quad \forall v\in V,\\&\sum_{a\in A}\beta_{ak}x_a \leq 1, \quad \forall k\in K,\\&x_a\in\{0,1\}, \quad \forall a\in A.\end{aligned} \tag{11}$$

In model 2, player 2 first delays the flight by attacking the legs, and then player 1 minimizes the total cost by adjusting the scheduling network. The flight delay probability of player 2's attack is 1, and the delay probability of other flights is determined by their relationship with the attacked flight. When all flights are not delayed, for any segment $v\in V$, the delay probability is $p_v = 0$, which is the same as the model in Section 2.1.

Assuming that there are multiple aircraft segments with delays and the two aircraft do not interfere with each other, we can limit each aircraft's flight path to a maximum of one aircraft that is delayed. Our model will change. So, we have the following linear programming (model 3):

$$\min_{g}\max_{V_1\subset V:||\leq|K|}\sum_{a\in A}c_a x_a + \sum_{v\in V}p_v c_v, \tag{12}$$

subject to

$$\begin{aligned}&\sum_{a\in A}\alpha_{av}x_a = 1, \quad \forall v\in V,\\&\sum_{a\in A}\beta_{ak}x_a \leq 1, \quad \forall k\in K,\\&x_a\in\{0,1\}, \quad \forall a\in A.\end{aligned} \tag{13}$$

In model 3, V_1 is a subset of V, such that for any two flight segments $v_1, v_2\in V_1$, v_1, and v_2 do not belong to any matching a at the same time. The delay probability is 1 if $v\in V_1$; otherwise, the delay probability is determined by the relationship with other flights.

3. Solution Approach

In model 2, there are $|A|$ decision variables, and the number of combinations of $|A|$ increases exponentially with the increase of flights. Also, there are $|V| + |K|$ constraints, the complexity and scale of the model are very large, so it is difficult to solve. A simple idea to solve this problem is to reduce the number of feasible schedules considered, since we can get an optimal schedule when no flight is delayed. Therefore, it is a feasible method to limit the scheduling cost to a certain proportion of the optimal scheduling. This makes sense, because in practical situations, a balance between cost and robustness is often pursued. For convenience, we refer to the two models above as the deterministic model (model 1) and the nondeterministic model (model 2 and model 3). Through the abovementioned description, if we want to solve the nondeterministic model, we can first solve the deterministic model, that is, to find the optimal value of the model in the absence of any flight segment delay.

3.1. Solution for Model 1. The branch pricing algorithm is an efficient algorithm for solving large-scale linear programming problems. The branch-pricing algorithm is a combination of a column-generating algorithm and a branch-and-bound algorithm. Among them, the column generation algorithm can be used to solve the solution of the relaxed model. The column generation algorithm narrows the range of candidate solutions and greatly reduces the amount of computation. After we use the column generation algorithm to obtain the candidate solution of the relaxed model, we can use the integer programming algorithm to solve it, but this often does not get the optimal solution to the original problem, so we need to use the column generation algorithm and the iteration of the branch and bound algorithm to solve it. However, in this paper, in order to save the time spent in the branch and bound process after obtaining the candidate solutions of the relaxed model; we start from the optimal solution of the integer programming and further obtain the suboptimal solution by using the variable neighborhood search algorithm. This greatly reduces the solution time of the model used.

3.1.1. Column Generation. We first linearly relax the aforementioned problem such that the variable $x_a\in[0,1]$, and call it the restrictive main problem (RMP). In general, not all matches will appear in the optimal solution, so we only need to consider adding some matches that reduce the objective function the most to the problem model, which is called the main problem of the column generation algorithm. In fact, only a small subset of all feasible matches will be added to the model. In this way, a solution can be found quickly, then look for variables outside the model, find variables that can make the model better, add the model to solve again, and so on until no better variables can be added to the model.

The way to find better variables is to find a solution that minimizes the objective function of the subproblem of the model. We set the dual variables corresponding to constraints 2 and 3 to be π_v and π_k, respectively. Then, the objective function of the subproblem is $\min Z = c_a - \sum_{v\in V}\alpha_{av}\pi_v - \sum_{k\in K}\beta_{ak}\pi_k = (c_p^k + c_k) - \sum_{v\in p}\pi_v - \pi_k = \sum_{v\in p}(c_v^k - \pi_v) + c_k - \pi_k$. Note here that, c_p^k represents the cost of path p allocated to aircraft k, and c_k represents the fixed cost of matching aircraft k contained in a, i.e., $c_a = c_p^k + c_k$, Therefore, we need to find a match that minimizes Z to add to the model. In this problem, although the number of planes is large, there may be only a few types of planes per airline. The cost for each type of aircraft to perform the same flight segment is the same, that is, for two aircraft of the same type $v_1, v_2\in V$ there are $c_{v_1}^k = c_{v_2}^k$. Therefore, we only need to find the matching k, p that minimizes Z. This corresponds to the shortest path problem with resource constraints, which can be solved by the labeling algorithm proposed by Si et al. [11]. We then add the resulting matches to the main problem until no match is found, such that $Z < 0$. In this way, we find all the candidate solutions corresponding to the relaxed model and then get the integer solution of the current candidate solution. But there is still a distance between the current solution and the optimal solution.

3.1.2. Variable Neighborhood Search Algorithm. Variable neighborhood search (VNS) [26] is an improved local search algorithm. It alternately searches using the neighborhood structure composed of different actions, achieving a good balance between concentration and evacuation. The VNS algorithm mainly relies on the selection of neighborhood sets in the Shaking phase. In the Shaking phase, the algorithm randomly generates a new solution from the kth neighborhood of the solution x. There are 4 commonly used neighborhood actions, namely: 2-opt [23], swap-move [24], insert [25], and exchange-move. But applied to this problem, not all operations are applicable. The algorithm designs three neighborhood actions, such as flight segment exchange, aircraft exchange, and insertion, and optimizes the solutions in sequence. Three kinds of neighborhood action search can deeply dig out the local optimal solution, and the specific methods are as follows:

Segment exchange: In the solution, two aircraft and segment path matches are randomly selected, and the segments in the two matched segment paths are swapped, as shown in Figure1. During this process, some exchanged solutions are infeasible, and we do not accept such infeasible solutions.

Aircraft exchange: In the solution, two matches are randomly selected, and the aircraft in them are directly exchanged to form a new solution, as shown in Figure 2.

Insert: In the solution, two matches are randomly selected, one or more flight segments in one of the matches are selected, and the selected flight segments are directly merged into the other match so that they are connected to form a new path, as shown in Figure 3.

Since it takes a lot of time to calculate in the branch and bound process, in order to improve the running speed, we use the variable neighborhood search algorithm in this part to get the optimal solution to the original problem. We combine the column generation algorithm and the variable neighborhood search algorithm to obtain algorithm. Steps 1–8 generate an initial solution for the column generation algorithm. Steps 9–31 improve the initial solution obtained by the column generation algorithm through domain search and finally obtain a suboptimal solution.

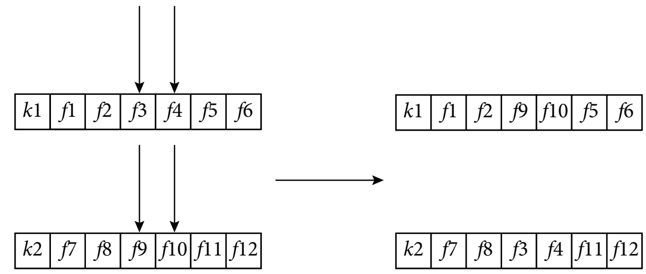

Figure 1: Segment exchange.

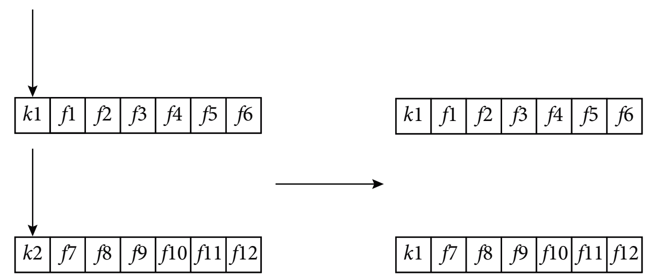

Figure 2: Aircraft exchange.

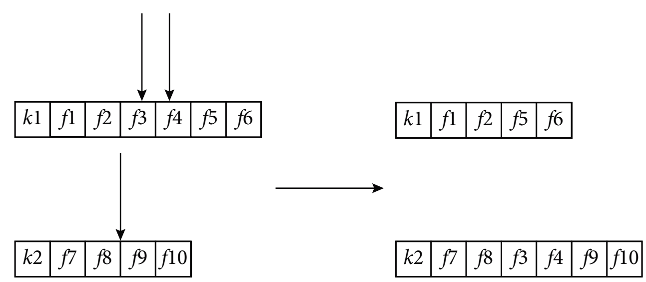

Figure 3: Insert.

3.2. Solution for Model 2. Due to the complexity and scale of model 2, it becomes difficult to solve, but in practical applications, we cannot only consider robustness but also robustness and cost. Therefore, we need to ensure that the generated scheduling cost does not exceed a certain percentage of the suboptimal solution cost generated in Section 3.1. To solve this problem, we developed an algorithm based on variable neighborhood search. The pseudocode for the variable neighborhood search algorithm applied to this problem is shown in algorithm. Note that F in algorithm represents the cost when there is no flight delay.

Step 1 is to generate an initial solution via algorithm. Step 2 is to calculate the worst-case cost of the initial solution. Step 3 is to initialize the number of iterations. Steps 4–27 are the main iterative process of the algorithm. Step 7 is the shake action. Steps 10–19 are the process of variable neighborhood descent. Steps 20–24 determine whether to update the current solution.

σ is a parameter that balances the two indicators of cost and stability. Since we describe the DARP problem as a minimization model, $\sigma \geq 1$. When $\sigma = 1$, it means that we only choose one schedule from the optimal schedule.

The algorithm is aimed at the situation where only one definite flight is delayed in the scheduling network. In practice, more than one flight is often delayed due to uncertainty. By making corresponding changes to the algorithm, the corresponding robust scheduling can be obtained. We can modify the single segment in algorithm to be a subset of all segments, so that we can get the solution for model 3.

3.3. Example. Here, we give a simple example to illustrate. The example contains 10 flight segments (for the sake of convenience, in the following description, Leg1 is written directly as $L1$), 2 aircraft of the same type. There is a certain fee for each aircraft to perform the corresponding flight segment. Also, there is a fixed fee for using the plane, which we set it as 2000. According to the China Civil Aviation Statistical Yearbook, the extra cost of one minute of delay for this type of aircraft is about 300, and the matching cost of the aircraft and flight segment is 20,000 per hour. In addition, other costs are not considered in this example. The flight segment information is shown in Table 1. We will use the game theory approach to solve the model where the utility function is a cost. Under the condition that all flight segments are not delayed, the optimal schedule of DARP can be obtained, as shown by n_1 in Table 2. Since there is only one type of aircraft, no distinction is made between aircraft for the time being. When the delay distribution of the flight segments is unknown, we can find the flight segment network g^* that satisfies the condition $F(g^*, v^*) = \min_g \max_v F(g, v)$, We can find other feasible schedules besides n_1 by setting $\sigma = 1.05$, as shown in Table 2. Table 3 lists the costs incurred by each solution when a flight segment is delayed. It can be seen from the cost table that there is no saddle point in this game, that is, there is no pure strategy Nash equilibrium. Nash equilibrium is a concept of solution in game theory. It refers to a combination of strategies that satisfy the following properties: any player who changes his strategy in this combination of strategies will not improve his own profits. We consider the more conservative case, $\min_{g\in G}\max_{v\in V} = 28.17$, with solutions n_4 and n_7 satisfying the condition. Hence, n_4 and n_7 are a robust schedule for this example.

4. Numerical Experiment

All experiments are run on an Intel(R) Core (TM) i7-1165G7 processor, 2.80 GHz, Windows 10 × 64 computer. The code of the algorithm uses the Python programming software and adopts the CPLEX commercial processor.

4.1. Problem Scenario and Dataset Information. Airline schedules often include multiple types of aircraft. In this setup, the airline needs to guarantee a reasonable connection time for each plane. In our experiments, the data provided by an airline in China is taken as an example to further illustrate the effectiveness of our proposed model and algorithm. For the aircraft model, the number of aircraft, the number of airports, the number of flight segments and other data included in the data, see (Table 4). As shown in Table 4, there are 7 types of aircraft in the dataset, including the "737," "73D," "73E," "73H," "73L," "73N," and "789." At the same time, more than 60 different airports are included in the dataset. Due to occasional aircraft maintenance, etc., the available number of aircraft in Table 4 may vary slightly. In the dataset, there are some canceled or planned canceled flights every day; in this case, we directly delete these canceled flights. Since the data of Wednesday is partially corrupted, we only use the dataset information of the remaining 6 days for the experiment. During the experiment, the dataset was divided into 6 independent datasets by date. Since our dataset contains the real scheduling information provided by airlines, it is itself a feasible solution for daily aircraft scheduling. Therefore, regardless of the airline's scheduling information, we use the basic information in the data for experimental evaluation.

4.2. Comparison of Branch Pricing Algorithm and Algorithm. In order to highlight the gap between the enumeration method and the column generation algorithm in solving large-scale linear programming, we first analyze the complexity of the two methods. For enumeration, it is necessary to find all feasible paths in the airline network. However, with the increase of flight segments, the scale of the deterministic model increases exponentially. On the contrary, for the column generation algorithm, only the paths that can reduce the objective function are found each time. The number of paths found by the column generation algorithm is far less than the number found by the enumeration method and can quickly approach the optimal. Therefore, it is infeasible to adopt complete enumeration for a large-scale aircraft routing problem.

First, we used the branch pricing algorithm and CG-VNS algorithm to test six instances, and the results are shown in Table 5. In the CG-VNS algorithm, the column generation time is shown in column CG-Time. In fact, CG-VNS and the branch pricing algorithm have the same column generation time. If we directly use CPLEX to solve all columns generated by the column generation algorithm, we can get an upper bound on the function value of the optimal integer solution. Finally, we list the gap between the CG-VNS solution and the branch pricing algorithm in the GAP column.

It can be seen from Table 5 that it is difficult to find large-scale examples of deterministic models using the branch pricing algorithm. The reason for the rapid increase in solution time is that the aircraft path in the connection network increases rapidly with the increase in instance size. This poses a great challenge to find high-quality solutions from a large number of decision variables. Furthermore, Table 6 reports the difference between the solution times of the VNS and CG-VNS algorithms and the solution time of the branch pricing algorithm. It can be seen from the calculation results in Table 6 that the calculation time of the VNS algorithm is lower than that of the CG-VNS algorithm. Compared with the branch pricing algorithm, the VNS algorithm is nearly 20 times faster than the standard branch pricing algorithm. However, the optimality gap of the VNS algorithm is far greater than that of the CG-VNS algorithm. The optimality gap of the CG-VNS algorithm is less than 0.1% except, for instance, on Friday. Another advantage of our CG-VNS algorithm is that it can quickly approach the optimal solution by taking a high-quality integer solution as the initial solution of the VNS algorithm.

In a word, our algorithm achieves a good balance between time and accuracy. It is better than the branch pricing algorithm in time and the VNS algorithm in precision. In other words, we sacrifice smaller function accuracy in exchange for greater improvement in runtime. This is obviously acceptable for airlines.

4.3. Comparison of Model 3 and Model 1 with Flight Delays. If a delay occurs, set the delay time to 30 minutes, and the cost per minute of delay is determined by the aircraft type. For the propagation probability of delay between two flight segments, if the transit time between the two flight segments is less than 60 minutes, the propagation probability is set to

```
Input: The maximum number of iterations Maxiter;
Output: Get a suboptimal solution x
    Use the DFS algorithm to generate an initial set O and initial path;
    while T ≠ ∅ do
        Using the Solver to find the optimal solution x of RMP and dual solutions π_v and π_k;
        Change arc costs in segment network using obtained dual solution;
        Find the matching T that minimizes Z by the labeling algorithm;
        Set O = O ∪ T
    end while
    Use the solver to find the integer solution x of RMP;
    t = 0;
    while t < Maxiter do
        k = 0;
        while k < k_max do
            Shaking: Randomly choose a solution x' from the k th neighbor N_k(x);
            VND: local search
            l = 0;
            while l < l_max do
                find a neighbor x'' in N_l(x');
                if F(x'') < F(x') then,
                    x' = x'', l = 0;
                else
                    l = l + 1;
                end if
            end while
            if F(x') < F(x) then,
                x = x', k = 0;
            else
                k = k + 1;
            end if
        end while
        t = t + 1;
    end while
```

ALGORITHM 1: Column generation and variable neighborhood search (CG-VNS).

0.8; if the layover time is 60 minutes to 120 minutes, the propagation probability is set to 0.3; if the layover time is greater than 120 minutes, the propagation probability is set to 0.005. At the same time, we assume that only one flight on each aircraft's route is delayed due to uncertainty factors and the other flights are delayed due to propagation. In this case, we will compare the cost of the solution produced by model 3 and the solution produced by model 1 in the case of flight delays. At the same time, we also compare the costs incurred by model 1 and model 3 when the flight is not delayed.

In Figure 4, it is visually shown that the cost of daily aircraft routing problems is different under different numbers of delayed flights and using different models. The abscissa in the figure corresponds to the number of flight segments that are assumed to be delayed; that is, when the abscissa is k, it corresponds to the objective function value of model 3 corresponding to the number of flights we assume to be k. The abscissa in Figure 4 corresponds to the number of flights assumed to be delayed. Assuming that the number of delayed flights is the green curve in Figure 4 corresponds to the worst-case cost of the scheduling generated by model 3 when the number of delayed flights is k. Obviously, when only one flight is delayed, that is, when $k = 1$, model 3 is equal to model 2. The blue curve in Figure 4 corresponds to the worst-case cost of the schedule generated by model 3 when the number of delayed flights is k. In Figure 5, it shows the cost of model 3 and model 1 when the user assumes that there are k flight delays but there are actually no flight delays. As can be seen from Figure 1, the cost incurred by model 3 is higher when there are actually no flight delays. Even so, we keep the cost below 1.05x of model 1. Combining Figures 4 and 5, we conclude that model 3 is more robust when flights are delayed due to uncertain factors. Even if none of the flights are actually delayed, the schedule generated by model 3 will not cost much.

4.4. Discussion. In conclusion, the abovementioned numerical results show that the uncertainty of flight delays will lead to an increase in airline costs. The reason for this is that it is impossible to know in advance which flights will be delayed during aircraft routing. Therefore, airlines can only assume that some segments will be delayed in advance to avoid some losses. However, before making this assumption, we also need to estimate the number of flights with delays. If the number is too large, the cost of the airline will increase more. If the number is too small, it will cause greater losses once the delay occurs. In fact, we can judge by experience. For example, if there is a typhoon in a certain area, there is a

Input: The maximum number of iterations Maxiter;
Output: Robust scheduling x; scheduling cost rob;

The solution x of model 1 is obtained by algorithm 1;
Compute rob $= \max_{v \in V} F(x, v)$;
$t = 0$;
while $t <$ Maxiter **do**
    $k = 0$;
    while $k < k_{\max}$ **do**
        Shaking: Randomly choose a solution x' from the kth neighbor $N_k(x)$;
        VND: local search
        $l = 0$;
        while $l < l_{\max}$ **do**
            find a neighbor x'' in $N_l(x')$;
            if $F(x'') < \sigma F(x')$ **then**
                if $\max_{v \in V} F(x, v) < \max_{v \in V} F(x', v)$ **then**
                    $x' = x'', l = 0$;
                else
                    $l = l + 1$;
                end if
            end if
        end while
        if $\max_{v \in V} F(x', v) <$ rob **then**
            rob $= \max_{v \in V} F(x, v), x = x', k = 0$;
        else
            $k = k + 1$;
        end if
    end while
    $t = t + 1$;
end while

Algorithm 2: Variable neighborhood search (VNS).

Table 1: Flight segment information table.

Leg	Airport dep	Time dep	Airport arr	Time arr
Leg1	Airport 1	03-29 12:05	Airport 2	03-29 14:00
Leg2	Airport 2	03-29 13:00	Airport 1	03-29 14:50
Leg3	Airport 1	03-29 18:05	Airport 3	03-29 19:15
Leg4	Airport 3	03-29 19:55	Airport 1	03-29 20:55
Leg5	Airport 1	03-29 21:55	Airport 3	03-29 23:15
Leg6	Airport 3	03-29 23:55	Airport 1	03-30 00:55
Leg7	Airport 1	03-29 08:05	Airport 6	03-29 09:10
Leg8	Airport 6	03-29 09:50	Airport 1	03-29 10:55
Leg9	Airport 1	03-29 08:05	Airport 3	03-29 09:15
Leg10	Airport 3	03-29 09:55	Airport 1	03-29 10:55

Table 2: Feasible scheduling.

Schedule	Route	Total costs (ten thousand)
n_1	$(L9, L10), (L7, L8, L1, L2, L3, L4, L5, L6)$	26.27
n_2	$(L7, L8), (L9, L10, L1, L2, L3, L4, L5, L6)$	26.27
n_3	$(L9, L10, L1, L2), (L7, L8, L3, L4, L5, L6)$	26.27
n_4	$(L9, L10, L3, L4), (L7, L8, L1, L2, L5, L6)$	26.27
n_5	$(L9, L10, L5, L6), (L7, L8, L1, L2, L3, L4)$	26.27
n_6	$(L7, L8, L5, L6), (L9, L10, L1, L2, L3, L4)$	26.27
n_7	$(L7, L8, L3, L4), (L9, L10, L1, L2, L5, L6)$	26.27
n_8	$(L7, L8, L1, L2), (L9, L10, L3, L4, L5, L6)$	26.27

Table 3: Costs when a flight is delayed.

	$L1$	$L2$	$L3$	$L4$	$L5$	$L6$	$L7$	$L8$	$L9$	$L10$
n_1	27.62	27.77	28.28	27.66	27.89	27.17	28.21	27.57	27.89	27.17
n_2	27.62	27.77	28.28	27.66	27.89	27.17	27.89	27.17	28.21	27.57
n_3	27.44	27.17	28.28	27.66	27.89	27.17	27.90	27.18	28.17	27.52
n_4	27.44	27.18	27.89	27.17	27.89	27.17	28.17	27.52	27.90	27.18
n_5	27.59	27.66	27.89	27.17	27.89	27.17	28.21	27.60	27.90	27.18
n_6	27.59	27.66	27.89	27.17	27.89	27.17	27.90	27.18	28.21	27.60
n_7	27.44	27.18	27.89	27.17	27.89	27.17	27.90	27.18	28.17	27.52
n_8	27.44	27.17	28.28	27.66	27.89	27.17	28.17	27.52	27.90	27.18

Table 4: Data information.

Instance	Aircraft type	Number of aircraft	Number of airports	Number of legs
Monday	7	83	63	350
Tuesday	7	83	63	318
Thursday	7	83	63	334
Friday	7	84	64	370
Saturday	7	84	63	329
Sunday	7	84	64	282

Table 5: CG-VNS vs BP.

Instance	CG-VNS				BP	
	CG-time (s)	VNS-time (s)	Total-time (s)	Gap (%)	Time (s)	GAP
Monday	278	33	311	0.05	727	—
Tuesday	235	35	270	0.02	724	—
Thursday	228	34	262	0.06	756	—
Friday	221	36	257	0.11	743	—
Saturday	253	32	285	0.08	703	—
Sunday	205	32	237	0.07	675	—

Table 6: VNS vs CG-VNS.

Instance	VNS		CG-VNS	
	Time (s)	Gap (%)	Time (s)	Gap
Monday	33	0.35	311	0.05
Tuesday	37	0.42	270	0.02
Thursday	39	0.46	262	0.06
Friday	34	0.51	257	0.11
Saturday	36	0.28	285	0.08
Sunday	31	0.17	237	0.07

high probability of a flight delay in that area. The deterministic model can provide a standard for the cost of the uncertain model, and the uncertain model can effectively cover the uncertainty in the display and reduce the loss caused by the uncertain factors. This is why we put forward the deterministic model and the uncertain model in this study. Therefore, the previous numerical experiments also verified that the mathematical model and optimization algorithm proposed by us can effectively absorb flight delays caused by uncertain factors.

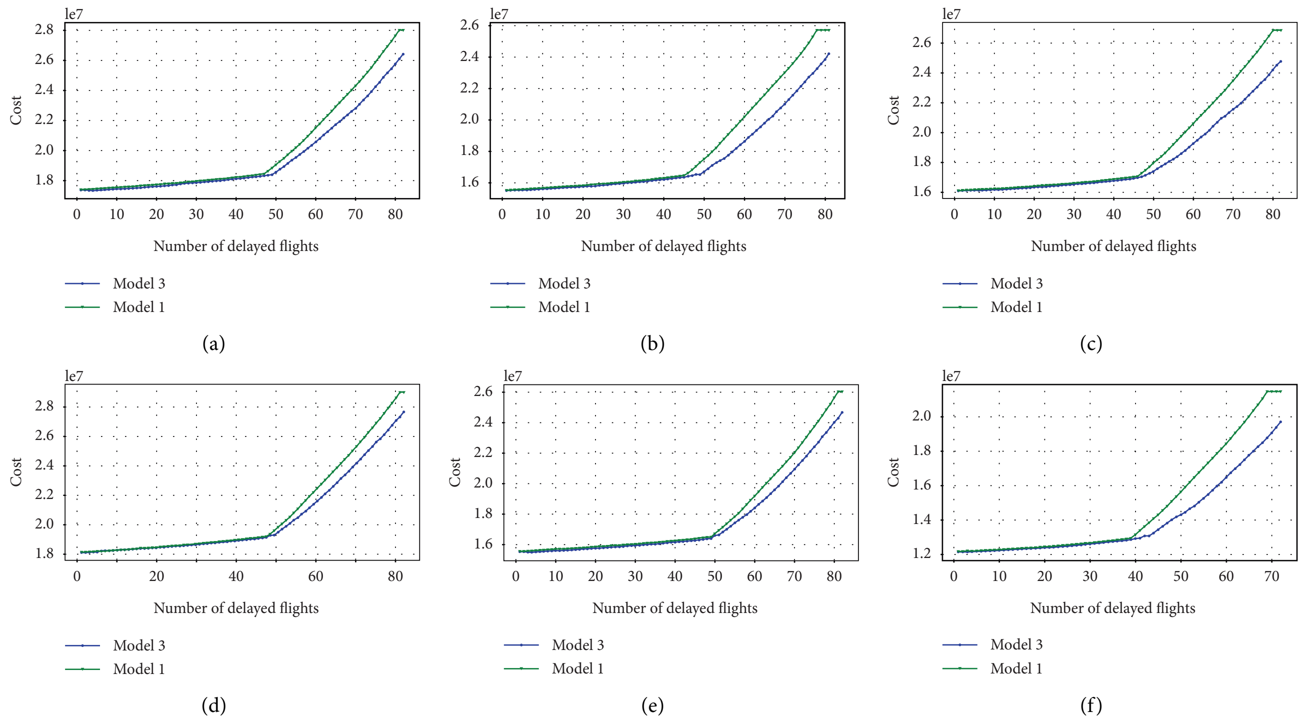

FIGURE 4: Delay costs under different models. (a) Monday. (b) Tuesday. (c) Thursday. (d) Friday. (e) Saturday. (f) Sunday.

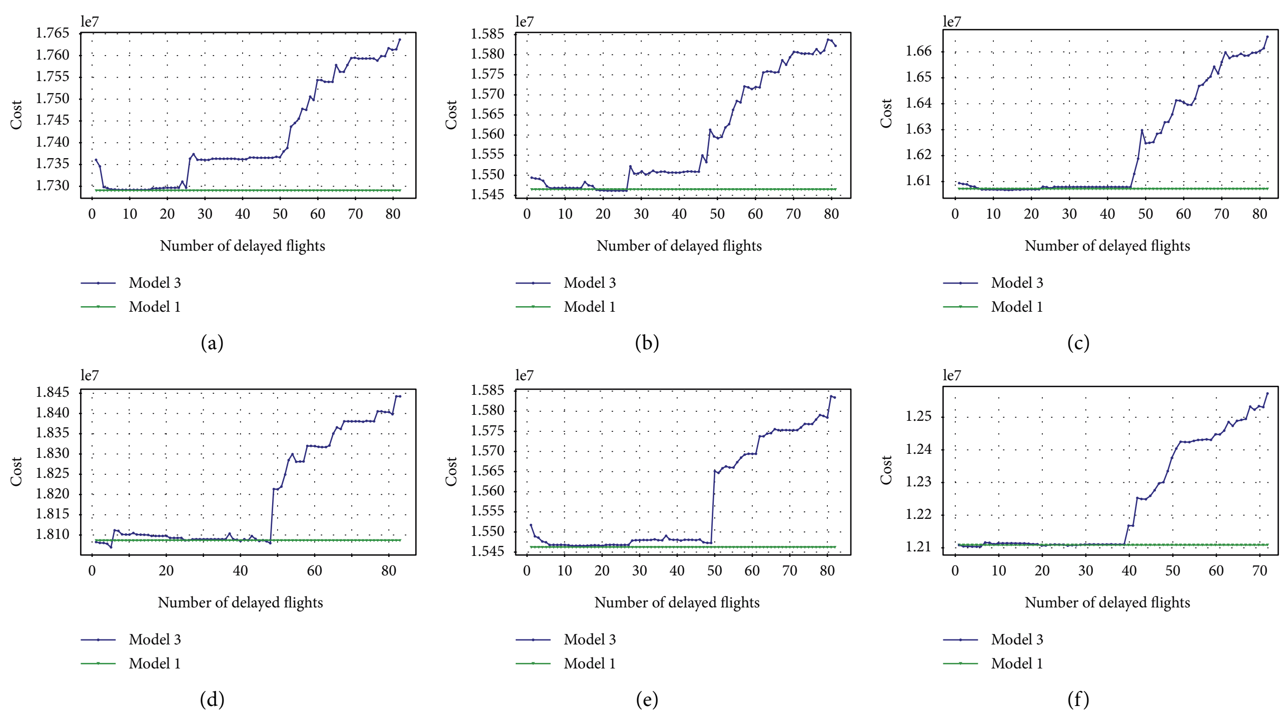

FIGURE 5: Nondelay costs under different models. (a) Monday. (b) Tuesday. (c) Thursday. (d) Friday. (e) Saturday. (f) Sunday.

5. Conclusion and Directions for Future Studies

The daily aircraft routing problem is one of the biggest challenges facing the aviation industry. For the daily aircraft routing and scheduling problem, we propose three mathematical models. The first model is a general, nonrobust model that aims to find a daily minimum cost for aircraft route scheduling. We build the second model based on game theory, which considers the flight propagation delay and the delay under uncertainty. As an extension of the second model, we established the third model. In the high-dimensional solution space, all three models are exponential models, so it is very difficult to effectively solve these models.

Therefore, for the general model, we also developed a VNS algorithm based on column generation. In addition, for the robust model, we also designed an improved VNS algorithm based on a column generation algorithm and proved that the proposed robust model can be effectively solved through reasonable calculations. On the one hand, a large number of real airline examples show that, compared with the exact solution generated by the branch pricing algorithm, this algorithm has a smaller gap and requires less time, so it has obvious advantages. On the other hand, compared with the nonrobust model, the robust model proposed by us has lower cost and stronger robustness.

In future research, we can consider, but not be limited to, the following directions. First, we can consider adding the impact of COVID-19 on flight delays to the model. Another interesting direction is to combine the pilot scheduling problem and the aircraft scheduling problem into a comprehensive problem because pilot scheduling [27] will also affect flight delays.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors' Contributions

Bin Deng proposed the innovation and first draft of this article; several other authors also contributed to the writing and revision of this article. Specifically, Jingfeng Li, Junfeng Huang, and Kaiyi Tang provided experimental data for the experiment in this paper, while Weidong Li and Hao Guo did a lot of work for the revision and finalization of this paper.

Acknowledgments

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

References

[1] A. Levin, "Scheduling and fleet routing models for transportation systems," *Transportation Science*, vol. 5, no. 3, pp. 232–255, 1971.

[2] G. Desaulniers, J. Desrosiers, Y. Dumas, M. M. Solomon, and F. Soumis, "Daily aircraft routing and scheduling," *Management Science*, vol. 43, no. 6, pp. 841–855, 1997.

[3] C. Barnhart, N. L. Boland, L. W. Clarke, E. L. Johnson, G. L. Nemhauser, and R. G. Shenoi, "Flight string models for aircraft fleeting and routing," *Transportation Science*, vol. 32, no. 3, pp. 208–220, 1998.

[4] D. Sosnowska, "Optimization of a simplified fleet assignment problem with metaheuristics: simulated annealing and GRASP," in *Approximation and Complexity in Numerical Optimization: Continuous and Discrete*, P. M. Panos, Ed., pp. 477–488, Kluwer Academic Publishers, Norwell, 2000.

[5] A. Mercier and F. Soumis, "An integrated aircraft routing, crew scheduling and flight retiming model," *Computers & Operations Research*, vol. 34, no. 8, pp. 2251–2265, 2007.

[6] O. Weide, D. Ryan, and M. Ehrgott, "An iterative approach to robust and integrated aircraft routing and crew scheduling," *Computers & Operations Research*, vol. 37, no. 5, pp. 833–844, 2010.

[7] A. Jamili, "A robust mathematical model and heuristic algorithms for integrated aircraft routing and scheduling, with consideration of fleet assignment problem," *Journal of Air Transport Management*, vol. 58, pp. 21–30, 2017.

[8] N. Kenan, A. Jebali, and A. Diabat, "An integrated flight scheduling and fleet assignment problem under uncertainty," *Computers & Operations Research*, vol. 100, pp. 333–342, 2018.

[9] L. Cadarso and R. de Celis, "Integrated airline planning: robust update of scheduling and fleet balancing under demand uncertainty," *Transportation Research Part C: Emerging Technologies*, vol. 81, pp. 3227–3245, 2017.

[10] R. Cui, X. Dong, and Y. Lin, "Models for aircraft maintenance routing problem with consideration of remaining time and robustness," *Computers & Industrial Engineering*, vol. 137, p. 106045, 2019.

[11] Y. Si, S. Gao, and W. Yang, "A new branch-and-price algorithm for daily aircraft routing and scheduling problem," in *International Conference on Algorithmic Applications in Management*, pp. 123–133, AAIM, 2021.

[12] Y. Xu, S. Wandelt, and X. Sun, "Airline integrated robust scheduling with a variable neighborhood search based heuristic," *Transportation Research Part B: Methodological*, vol. 149, pp. 181–203, 2021.

[13] O. Şafak, O. Çavuş, and M. S. Aktürk, "A two-stage decision dependent stochastic approach for airline flight network expansion," *Transportation Research Part B: Methodological*, vol. 158, pp. 78–101, 2022.

[14] A. Goli and H. Mohammadi, "Developing a sustainable operational management system using hybrid Shapley value and Multimoora method: case study petrochemical supply chain," *Environment, Development and Sustainability*, vol. 24, no. 9, pp. 10540–10569, 2022.

[15] D. A. Lima, J. Contreras, and A. Padilha-Feltrin, "A cooperative game theory analysis for transmission loss allocation," *Electric Power Systems Research*, vol. 78, no. 2, pp. 264–275, 2008.

[16] N. Jia and R. Yokoyama, "Profit allocation of independent power producers based on cooperative Game theory," *International Journal of Electrical Power & Energy Systems*, vol. 25, no. 8, pp. 633–641, 2003.

[17] S. Mei, D. Zhang, Y. Wang, F. Liu, and W. Wei, "Robust optimization of static reserve planning with large-scale integration of wind power: a game theoretic approach," *IEEE Transactions on Sustainable Energy*, vol. 5, no. 2, pp. 535–545, 2014.

[18] G. Laporte, J. A. Mesa, and F. Perea, "A game theoretic framework for the robust railway transit network design problem," *Transportation Research Part B: Methodological*, vol. 44, no. 4, pp. 447–459, 2010.

[19] F. Perea and J. Puerto, "Revisiting a game theoretic framework for the robust railway network design against intentional

attacks," *European Journal of Operational Research*, vol. 226, no. 2, pp. 286–292, 2013.

[20] M. Alinaghian and A. Goli, "Location, allocation and routing of temporary health centers in rural areas in crisis, solved by improved harmony search algorithm," *International Journal of Computational Intelligence Systems*, vol. 10, no. 1, pp. 894–913, 2017.

[21] Q. Yang, J. Wu, Y. Li, W. Li, L. Wang, and 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," *Agricultural and Forest Meteorology*, vol. 232, pp. 606–622, 2017.

[22] Q. Yang, L. Dan, J. Wu et al., "The improved freeze–thaw process of a climate-vegetation model: calibration and validation tests in the source region of the Yellow River," *Journal of Geophysical Research: Atmospheres*, vol. 123, no. 23, pp. 13–346, 2018.

[23] G. A. Croes, "A method for solving traveling-salesman problems," *Operations Research*, vol. 6, no. 6, pp. 791–812, 1958.

[24] I. M. Oliver, D. J. Smith, and J. R. C. Holland, "Study of permutation crossover operators on the traveling salesman problem," *Proceedings of the second International Conference on Genetic Algorithms*, vol. 28, pp. 477–488, Cambridge, MA, 1987.

[25] D. B. Fogel, "An evolutionary approach to the traveling salesman problem," *Biological Cybernetics*, vol. 60, no. 2, pp. 139–144, 1988.

[26] N. Mladenović and P. Hansen, "Variable neighborhood search," *Computers & Operations Research*, vol. 24, no. 11, pp. 1097–1100, 1997.

[27] B. Deng, R. Ding, J. Li, J. Huang, K. Tang, and W. Li, "Hybrid multi-objective metaheuristic algorithms for solving airline crew rostering problem with qualification and language," *Mathematical Biosciences and Engineering*, vol. 20, no. 1, pp. 1460–1487, 2022.