Mathematical Biosciences and Engineering

MBE, 20(7): 12802–12819.
DOI: 10.3934/mbe.2023571
Received: 01 April 2023
Revised: 11 May 2023
Accepted: 15 May 2023
Published: 31 May 2023

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

Research article

Aircraft route recovery based on distributed integer programming method

Luyao Yang, Zhikang Wang, Haochen Yu, Baoping Jiang and Zhengtian Wu*

School of Electronic and Information Engineering, Suzhou University of Science and Technology, Suzhou, China

* **Correspondence:** Email: wzht8@mail.usts.edu.cn.

Abstract: In order to further promote the application and development of unmanned aviation in the manned field, and reduce the difficulty that airlines cannot avoid due to unexpected factors such as bad weather, aircraft failure, and so on, the problem of restoring aircraft routes has been studied. To reduce the economic losses caused by flight interruption, this paper divides the repair problem of aircraft operation plans into two sub problems, namely, the generation of flight routes and the reallocation of aircraft. Firstly, the existing fixed-point iteration method proposed by Dang is used to solve the feasible route generation model based on integer programming. To calculate quickly and efficiently, a segmentation method that divides the solution space into mutually independent segments is proposed as the premise of distributed computing. The feasible route is then allocated to the available aircraft to repair the flight plan. The experimental results of two examples of aircraft fault grounding and airport closure show that the method proposed in this paper is effective for aircraft route restoration.

Keywords: flight interruption; fixed-point iteration; integer programming; distributed computing

1. Introduction

At present, civil pilotless aviation is developing rapidly and iteratively. Unmanned aviation has gradually become an important carrier of advanced productivity. Owing to its convenience and time-saving characteristics, air travel has become the preferred way for people to travel nowadays, especially for long-distance travel. The standard for continuous and on-time flight plans should be high because if the flight is not punctual, it will affect the travel plans of passengers, bring dissatisfaction to passengers and cause economic losses to airlines. The flight plan operated by airlines is usually made up of four parts, namely, flight arrangement, fleet allocation, aircraft route, and crew arrangement. The development of a flight schedule model is the best operation plan for airlines [1]. Markov chains [2] can be used to model the bad weather that affects flight delays. Some flights are often interrupted by uncontrollable factors such as aircraft maintenance, bad weather, and unavailability of crew; consequently, the original plan cannot be implemented smoothly. Flight interruption or airport closure will seriously affect subsequent flights and airport operation plans, causing a large number of flights to be delayed or even canceled [3]. Airlines must immediately make a decision to reallocate available aircraft and reschedule flight schedules by formulating recovery plans to reduce airline flight cancellation losses.

Route restoration is always an NP-hard optimization problem [4]. Teodorovic and Gubernic [5] were the first to study abnormal flight scheduling. They established a mathematical model to minimize passenger delay, adjust the abnormal original route network and use the branch and bound method to solve the established network model. Jarrah [6] proposed a route disturbance caused by the short-term grounding of aircraft in Beijing and established a network flow model aiming at minimizing the cost of flight delays and cancellations. The limitation of this model is that it can only be used alone for flight delays or cancellations. Cao and Kanafani [7] improved Jarrah's abnormal flight scheduling model which cannot comprehensively consider flight delays and flight cancellations. They established a 0–1 quadratic programming model with the goal of maximizing profits and extended the model to special environments such as different models. Zhou [8] established a mathematical model for minimizing the total delay time of passengers and improved the genetic algorithm to restore the abnormal flight schedule. After understanding the NP problem of aircraft recovery, Wang [9] proposed the greedy random adaptive search algorithm grasp algorithm and verified the feasibility of this algorithm in solving the aircraft recovery problem. Through comparative experiments, the limitations of the Lagrange relaxation algorithm in rescheduling were revealed. Lin [10] proposed an fast variable neighborhood search (FVNS) algorithm composed of construction and improvement stages to solve the problem of flight disturbance caused by airport closure. The algorithm is based on fast variable domain search. In the construction stage, flights affected by unexpected factors are considered to delay their departure time and generate feasible routes. In the first stage of improvement, the goal is to minimize the delay cost, find exchangeable flights and then obtain the best flight schedule. Kenan [11] developed three different column generation algorithms, including fleet allocation, flight schedule and aircraft route, to solve the interruption problem. To solve the problem of a large number of route disturbances caused by the temporary closure of the airport, Lin [12] made a new definition of flight recovery, simplified the flight plan with complex constraints by using sequential decision-making and prioritized the aircraft with higher losses to enter the flight plan. The local search heuristic method is the main method to solve the problem of route restoration. With the help of a local search heuristic algorithm [13,14], this problem can be handled with higher efficiency. Reassigning aircraft to resume aircraft routes is a combinatorial optimization problem, which can be effectively solved by using a deterministic annealing neural network algorithm [15]. The deterministic annealing neural network algorithm also has good applications in production transportation [16] and cost transportation [17]. Fogaça [18] monitored the decision making during the flight interruption through the operation control center and determined five mechanisms to eliminate the interference to the flight plan caused by the interruption. Lee [19] considered the weather change factors affecting flight recovery and built a rolling horizon model to deal with the recovery of integrated aircraft and passengers. Bouarfa [20] proposed a multi-agent system method to evaluate the performance of airline operation control and recover the interrupted flight plan. Therefore, the models and algorithms built in the above documents are based on the overall solution to the flight recovery problem. The main contribution and innovation of this paper is to divide the flight recovery problem into two steps, namely, feasible route generation and aircraft reassignment, and propose a distributed computing method based on fixed point theory to solve the flight recovery problem. First, with the help of fixed point theory, feasible routes can be obtained in lexicographic order, facilitating the generation of new operation plans. A second point is that large-scale and complex problems can be divided into several sub-problems to simplify calculation, which can save time and improve accuracy.

The remainder of this paper is structured as follows. In Section 2, the two sub problems of aircraft route restoration are summarized and modeled, respectively. In Section 3, the distributed computing network is introduced, and a segmentation method is proposed to divide the problem space into several independent segments. In Section 4, the performances of the distributed Dang method and CPLEX in solving the generation of feasible routes are compared through numerical examples. The results of aircraft route recovery in the case of aircraft failure and temporary airport closure are given. In Section 5, the work of this paper is summarized.

2. Problem description and modeling

Owing to some bad weather conditions such as typhoons or rainstorms or aircraft maintenance failure, some aircraft cannot fly according to the original plan within a certain period. The main content of this paper is how to maximize the benefits of using the remaining available aircraft to solve this kind of interruption problem. In case of interruption, if measures are not taken in time to respond positively, the flight planned by the affected aircraft is likely to be canceled. To greatly reduce the economic losses caused by the interruption to airlines, the available aircraft should be allowed to undertake the flight tasks of unavailable aircraft as much as possible so that a new flight plan can be obtained. This interruption problem can be divided into two sub problems. One is to generate all feasible flight routes of available aircraft according to the original flight plan, and the other is to reallocate feasible flight routes to the remaining available aircraft to minimize losses.

2.1. Generate feasible routes

The feasibility of this sub problem is further described through the airport connection network in Figure 1. The circular nodes marked by HF, SH and BJ in the figure represent three airports. The arcs marked by numbers represent the flights between the two airports. Two source and sink nodes are added on the left and right sides to connect the airport nodes and the flight arc to show the starting and ending points of each route. As there is no route to take off or land from the BJ node, there is no arc between this node and source and sink nodes. The network connection needs to meet the entry-exit balance condition. That is, the number of arcs flowing out of the airport node should be equal to the number of arcs flowing into the airport node. However, the limitation of this network diagram is that it cannot arrange the flight sequence of flights, so we introduce heuristic transformation to effectively solve this limitation problem and determine the flight sequence of each segment of the flight route.

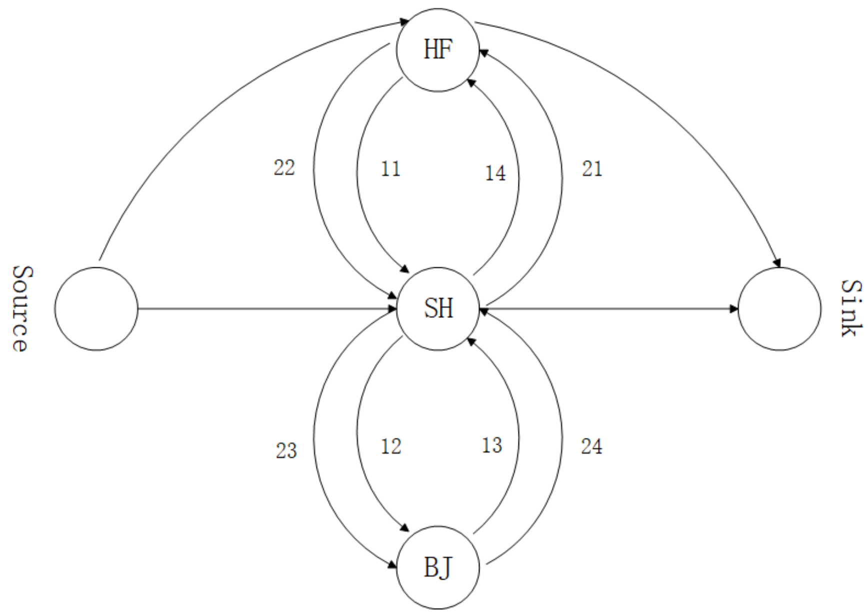

Figure 1. Airport connection network.

To find the arc combination that meets various constraints in the network diagram, the following model is established. The model cannot determine the flight sequence, so the solution process is relatively simple.

The following is a description of the symbols in the model.

Subscript:

i : flight subscript

p : airport subscript

Set:

F : flights

S : airports

Parameters:

d_{ti} : duration of flight i

t_{ti} : turnaround time of flight i

$$s_{ip} = \begin{cases} 1 & flight\ i\ departs\ from\ airport\ p \\ 0 & otherwise \end{cases}$$

$$a_{ip} = \begin{cases} 1 & flight\ i\ arrives\ at\ airport\ p \\ 0 & otherwise \end{cases}$$

Decision variables:

$$x_i = \begin{cases} 1 & flight\ i\ included\ on\ feasible\ routes \\ 0 & otherwise \end{cases}$$

$$o_p = \begin{cases} 1 & the\ departure\ airport\ of\ feasible\ route\ is\ p \\ 0 & otherwise \end{cases}$$

$$f_p = \begin{cases} 1 & the\ finishing\ airport\ of\ the\ feasible\ route\ is\ p \\ 0 & otherwise \end{cases}$$

Mathematical model of feasible route:

$$\exists x_i, o_p, f_p \in \{0,1\}, \forall i \in F, \forall p \in S \tag{1}$$

$$-\sum_{i\in F} x_i s_{ip} + \sum_{i\in F} x_i a_{ip} + o_p - f_p = 0, \forall p \in S \tag{2}$$

$$o_p - \sum_{i\in F} x_i s_{ip} \le 0, \forall p \in S \tag{3}$$

$$f_p - \sum_{i\in F} x_i a_{ip} \le 0, \forall p \in S \tag{4}$$

$$\sum_{p\in S} o_p = 1 \tag{5}$$

$$\sum_{p\in S} f_p = 1 \tag{6}$$

where x_i is the flight sequence arranged in ascending order according to the flight take-off time. Parameters d_{ti} and t_{ti} and sets F and S can be easily obtained from the original flight schedule. Parameters o_p and f_p can also be obtained from the starting and ending points of each leg in the observation schedule.

This model is to calculate all feasible flight routes that meet constraints (2)–(6). Constraint (2) requires that the number of outbound flights at each airport is equal to the number of inbound flights. Constraints (3) and (4) require that the departure airport and termination airport of a feasible route must be the airport pointed by the source node and sink node. Constraints (5) and (6) require that each route has only one departure airport and one termination airport.

To convert the numbers of 0 and 1 calculated by the above model into the actual flight route, the steps of heuristic algorithm conversion are introduced below.

Step1: From the above calculation results, we can obtain the s_{ip} of the departure airport, the a_{ip} of the termination airport and the selected flight segment x_i of the feasible route.

Step 2: We put the flights with the same origin airport in the solution into one storage package. If there are multiple flights with the same origin airport, we put these flights into the storage package.

Step 3: We store a single flight that has not been stored in the storage package and whose departure airport is the same as the destination airport of the last flight in the package in the end position of the storage package. If there are multiple flights that have not been stored, we will copy the storage package for these flights that have not been stored, so that each flight belongs to a storage package.

Step 4: We repeat steps 2 and 3 to store all the selected feasible flights in the corresponding storage package.

Step 5: We delete the storage package whose capacity is less than the number of flights selected in the solution, and then we keep the storage package of the route with the shortest delay time.

Step 6: We obtain the actual and feasible routes after sorting.

```
Input: {s_ip, a_ip, x_i}
Output: A feasible route
begin
for each p with s_ip = 1 do
    for each i do
    if(x_i = 1 && flight_i.origin = station_p)
        begin
        Store flight_i in the vector container Origin_flight;
        end
for each p with a_ip = 1 do
    for each i do
    if(x_i = 1 && flight_i.destination = station_p)
        begin
        Store flight_i in the vector container Destination_flight;
        end
Generate |Origin_flight| multiple guest Flight_route to store each flight_i in the
Origin_flight;
for each flight_route do
    begin
    Generate vector container flights_to_be_added;
    for each i do
        begin
        If(x_i = 1 && flight_i ∉ Flight_route && the arrival flight of the last flight in vector
    container Flight_route = flight_i.origin)
            begin
            Store flight_i in flight_to_be_added;
            end
            Copy the vector container Flight_route (|flights_to_be_added| − 1)
        times, and then store each flight_i in the vector container flights_to_be_added
        separately in Flight_route and the copied copy;
            end
        end
    Store all Flight_route and their copies in a vector container Flight_route_list;
    for each feasible route in Flight_route_list do
        Delete all other routes in the Flight_route_list except for the one with the least
    delay time;
    end
end
```

Figure 2. Heuristic transformation of pseudocode.

In the mathematical model for generating feasible routes, as shown in Figure 1, the number of aircraft arriving at the airport should be equal to the number of aircraft departing from that airport. The original flight plan of the flight should be used as the initial solution, such as the takeoff and landing times of the aircraft, as well as the takeoff and arrival at the airport. The feasible solution should be calculated using a fixed point iterative algorithm, and then transformed into a practical feasible path through a heuristic algorithm.

2.2. Reassign aircraft

When the feasible route generation problem is solved, the feasible route set is generated, and the second sub problem of flight interruption can be solved by modeling as a resource assignment problem [21]. The resource assignment problem is a network path flow model which requires the feasible flight routes be arranged to the available aircraft resources within certain constraints and that the objective function be minimized. However, solving the resource assignment problem is difficult. In the face of large-scale problems, there may be no way to completely calculate the feasible route. The distributed integer programming method proposed in this paper can effectively solve this kind of problem. This method does not need to calculate all the feasible routes, and only a part of it is needed to realize aircraft reassignment.

The following is a description of the symbols in the model.

Subscript:

i : flight subscript

j : route subscript

p : airport subscript

Set:

F : flight assembly

S : airport assembly

R : feasible route set

Parameters:

dl_j : loss caused by delayed route j

cl_i : loss caused by flight cancellation i

u_p : number of planes that need to take off from airport p before dispatching

v_p : the number of aircraft that need to stay at airport p at the end of scheduling

sn : total number of aircraft

$$x_{ij} = \begin{cases} 1 & \text{flight } i \text{ is in route } j \\ 0 & \text{otherwise} \end{cases}$$

$$o_{pj} = \begin{cases} 1 & \text{route } j \text{ starts from airport } p \\ 0 & \text{otherwise} \end{cases}$$

$$f_{pj} = \begin{cases} 1 & \text{route } j \text{ ends at airport } p \\ 0 & \text{otherwise} \end{cases}$$

Variables:

$$y_j = \begin{cases} 1 & \text{assign aircraft to execute route } j \\ 0 & \text{otherwise} \end{cases}$$

$$k_i = \begin{cases} 1 & \text{cancel flight } i \\ 0 & \text{otherwise} \end{cases}$$

Aircraft allocation model:

Minimize

$$\sum_{j\in P} dl_j y_j + \sum_{i\in F} cl_i k_i \tag{7}$$

subject to:

$$\sum_{j\in P} x_{ij} y_j + k_i = 1, \forall i \in F \tag{8}$$

$$\sum_{j\in P} o_{pj} y_j = u_p, \forall p \in S \tag{9}$$

$$\sum_{j\in P} f_{pj} y_j = v_p, \forall p \in S \tag{10}$$

$$\sum_{j\in P} y_j = sn \tag{11}$$

$$y_j, k_i \in \{0,1\}, \forall j \in P, \forall i \in F \tag{12}$$

Among them, parameters x_{ij}, o_{pj} and f_{pj} are the solutions calculated in the feasible route model, and the remaining parameters can be obtained by observing the flight schedule. Objective function (7) requires the minimum total loss caused by route delay and flight cancellation. Constraint (8) requires that there are only two situations for any flight, either assigned to a feasible route or cancelled. Constraints (9) and (10) require that the number of aircraft taking off from airport *P* and the number of aircraft finally landing at airport P reach a balance before and after flight interruption. Constraint (11) requires that the number of aircraft that can be assigned tasks is equal to the number of feasible routes finally selected. Constraint (12) specifies that the value of a variable can only be 0 or 1.

3. Distributed method

This section uses the segmentation method introduced in [22] to divide the geometric space generated by model 2.1, and all sub segments after division have no intersection. With this feature, all sub segments can be calculated by independent processors. In this paper, two dual core computers and a four-core computer are used in the experiment, and the distributed computing toolbox in MATLAB is used to realize distributed parallel computing. The flow chart of distributed parallel computing is shown in Figure 3.

We start a service called MATLAB distributed computing engine in each computer participating in the calculation, and we use this service to start the MATLAB session of the workers participating in the calculation and the job manager managing the workers of each computer. The job manager manages workers, assigns calculation tasks to workers and receives the calculated results of workers. In the host computer, that is, the client, the problem is divided into several segments by the segmentation method in [22], and the starting point x_i^s and ending point x_i^e of each sub segment are sent to the job manager. The job manager then allocates tasks according to the number and status of workers. When the number of sub segments allocated is the same as the number of workers, these tasks can be calculated at the same time. If the number of sub segments allocated is greater than the number of workers, the workers who first terminate the operation can be used to solve the remaining sub segments.

When the task of a computer is completed, all the solutions obtained can be sent to the host computer. In short, the purpose of distributed computing is to divide the complex work into several small tasks, expand it from one computer to multiple computers and finally summarize the calculation results so as to improve the calculation efficiency.

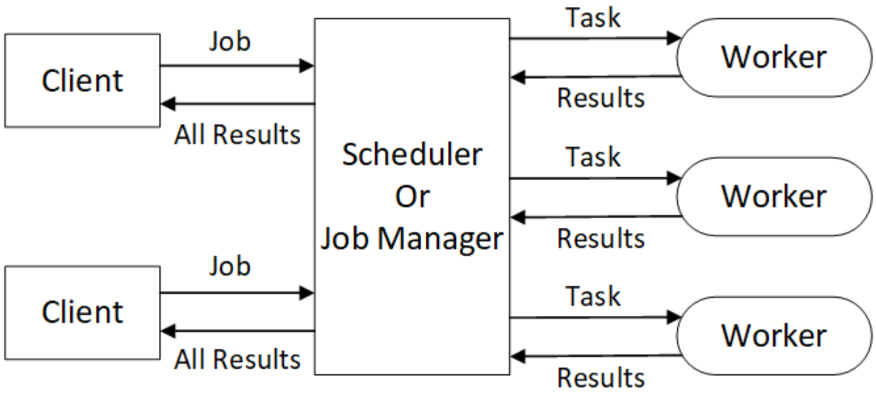

Figure 3. Interaction of distributed computing.

The problem of feasible route generation can be transformed into the calculation problem of integer points in a polyhedron with the help of the integer programming iteration method proposed by Dang [23,24]. The design idea of the algorithm is to define an incremental mapping from the problem lattice to the lattice itself based on the fixed-point theory and the incremental mapping principle. The mapping rules are designed based on the lexicographical order mapping principle, so that the integer point outside the polyhedron is mapped to a feasible integer point inside the polyhedron that is smaller than the lexicographical order of the integer point. The operation is carried out according to the iterative steps. If the feasible solution is not met, return to the initial step and carry out the next iteration. When no feasible solution is found within the polyhedron, the algorithm is terminated. According to the dictionary order, all integer points in the polyhedron can be connected into a one-dimensional sequence, which can be divided into multiple sub segments. The segments do not affect one another, which meets the requirements of distributed computing, that is, multiple workers can be used to calculate each sub segment at the same time.

The segmentation method used in this paper is to divide the problem space evenly, that is, the number of integer points contained in each sub segment formed by the space after segmentation is basically equal. When facing the relatively large-scale aircraft flight interruption problem, to effectively repair the flight plan, a sufficient number of workers must be prepared to divide the problem space into several relatively small sub segments.

Figure 4 is a two-dimensional example showing the average segmentation method. Firstly, the constraint problem is transformed into the constraint triangle in the figure. All integer points in the problem space are represented by horizontal and vertical coordinates to form an integer lattice. We use $R(t)$ to represent the integer lattice, and S is the number of sub segments to be divided. If all integer points in $R(t)$ can be evenly divided into S sub segments, the number of integer points in each sub segment is $\frac{|R(t)|}{S}$. If not, we divide $\left\lfloor \frac{|R(t)|}{S} + 1 \right\rfloor$ integer points in the $\mathrm{mod}(|R(t)|, S)$ sub segment for calculation and divide $\left\lfloor \frac{|R(t)|}{S} \right\rfloor$ integer points in the remaining sub segment for calculation. The integer points contained in each sub segment of the segmentation result obtained in this way are either equal or have only one difference. The sum of the integer points in all sub segments is equal to the number of integer points in the integer lattice $R(t)$, which can be verified by formula (13).

$$\begin{aligned} &\bmod(|R(t)|, S) \times \left\lfloor \frac{|R(t)|}{S} + 1 \right\rfloor + (S - \bmod(|R(t)|, S)) \times \left\lfloor \frac{|R(t)|}{S} \right\rfloor \\ &= S \times \left\lfloor \frac{|R(t)|}{S} \right\rfloor + \bmod(|R(t)|, S) \\ &= |R(t)| \end{aligned} \tag{13}$$

The integer lattice $R(t)$ in Figure 3 is divided into four sub segments. One of which is divided into 39 integer points, and the other three sub segments are divided into 38 integer points. The sum of the integer points of the four sub segments is 39 + 38 × 3 = 153, that is, the total number of integer points in the integer lattice $R(t)$. Here, black squares and blue pentagons are used to represent integer points in adjacent segments. The challenge of segment segmentation is to calculate the start x_i^s and end x_i^e of each segment. Using formula (13), the values of x_i^s and x_i^e can be determined quickly and effectively.

$$x_i^s = \begin{cases} x_i^u - \left\lfloor \dfrac{\bmod\left(d, \prod_{j=i}^{n} c_j\right)}{\prod_{j=i+1}^{n} c_j} \right\rfloor & i = 1,2,\ldots,n-1 \\ x_n^u - \bmod(d, c_n) & i = n \end{cases} \tag{14}$$

where x^u is the minimum and maximum value of the constraint region, d is the number of integer points between the partition point to be calculated and x^u in the dictionary order, c_j is the number of integer points in the j-dimensional coordinate, and N is the dimension of the problem space. x_i^e can also be calculated by formula (14), but we pay attention to the size change of d. After the start and end points of each sub segment are obtained, the feasible solutions in the corresponding segment can be solved using the distributed computing structure.

In distributed computing, the problem space is evenly divided into several parts based on the proposed average segmentation method. The initial point for each processor in problem-solving is the point assigned to the processor with the fewest dictionary-ordered segments. Then, according to the fixed-point iterative algorithm, all feasible solutions in each segment are solved, that is, the output items of each processor and each process are summarized and sent to the host to obtain feasible solutions for the entire problem space.

Building a distributed computing framework based on the Integer programming algorithm not only improves the computational efficiency of the algorithm in theory but also makes full use of computing resources and reduces the time cost. In the practical application of solving flight interruption, it can enable rapid response to emergency events and reduce unnecessary losses for airlines.

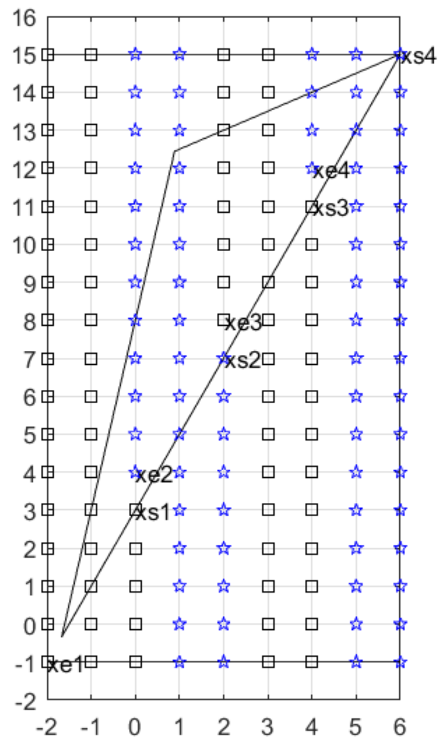

Figure 4. Average segmentation method.

4. Example analysis

Based on the above-mentioned results, the model algorithms used in this study were programmed by MATLAB. We built a distributed parallel computing environment with three operating systems for Windows 10 (64 bits), two dual-core laptops and one quad-core laptop. The code is written in MATLAB 2016b.

In this paper, the effectiveness of the proposed recovery interruption model is verified by a certain airline's partial flight operation plan on a certain day. In case of no emergency, the aircraft will operate according to the original plan, and the specific flight schedule is shown in Table 1.

According to the original flight schedule, this example involves 11 aircraft, 20 flights and 18 airports. This chapter first considers the situation that the sudden mechanical failure of the aircraft A8 results in the inability to fly normally throughout the day. In the event of such an emergency, the airline's most direct decision is generally to cancel all flight missions originally performed by the aircraft A8. Although this is the most time-saving, the economic loss to the airline cannot be underestimated. If the airline does not consider rescheduling and directly cancels the three flights of the aircraft A8, it will cause 30,551 in economic losses to the airline. However, if time and other conditions allow, the remaining aircraft that can normally perform the flight mission can be used to perform the flight mission of aircraft A8 as far as possible, and the purpose of reducing economic losses can be achieved. Here, $delayCost(t) = 30t$ (t is minutes) is used as the delay cost of each flight to participate in the calculation, that is, if the flight is delayed for 1 minute, a loss of 30 yuan is incurred, and the minimum stop time of the aircraft is 40 minutes.

The original flight schedule given in Table 1 can obtain all feasible routes within the effective time under the distributed computing structure introduced in Section 3. However, owing to the different hardware performances of the three computers used in this experiment, the numbers of solutions generated by the sub segments formed by the average segmentation of the feasible route solution space in each computer is significantly different. The generation of feasible routes after segmentation is combined with Dang's fixed-point iterative algorithm. The results are shown in Table 2.

Table 1. Schedule of normal flight operation.

Aircraft	Flight	Origin	Destination	Departure	Arrival	Cancellation loss
A1	11	CKG	NKG	11:05	13:00	13209
	12	NKG	INC	13:50	16:10	16862
A2	21	TAO	ICN	13:45	14:50	14354
	22	ICN	TAO	15:40	17:00	12586
A3	31	ZGC	JGN	14:40	15:50	9583
A4	41	TAO	FUK	11:05	12:40	6864
	42	FUK	TAO	13:40	16:00	15349
A5	51	KWE	KHN	11:45	13:10	14341
	52	KHN	NKG	13:55	15:00	10183
A6	61	NKG	BAV	13:00	15:00	12524
	62	BAV	NKG	15:50	17:40	9352
A7	71	HGH	CAN	14:05	16:10	9961
A8	81	PEK	NKG	10:45	12:40	10263
	82	NKG	CSX	13:30	14:55	8465
	83	CSX	PEK	15:45	17:45	11823
A9	91	HRB	TAO	12:00	13:55	12642
	92	TAO	NKG	14:45	15:55	8746
A10	101	LZO	PEK	10:00	11:10	7284
	102	PEK	LZO	11:55	15:30	10970
A11	111	PVG	DAT	15:45	18:35	15646

Table 2. Performance of each worker using the average segmentation method.

	Worker	Number of Solutions	Total solution time/ms	Average time of solution/ms
Computer A	1	42	7848	186.86
	2	29	5567	191.97
Computer B	1	141	32963	233.78
	2	16	931	58.19
Computer C	1	368	56357	153.14
	2	93	26352	283.35
	3	496	52674	106.2
	4	1067	263955	247.38

The data in Table 2 reveal that the original complex feasible route generation problem is evenly distributed to three computers for calculation, two of which contain two workers, and one computer contains four workers. Therefore, the eight sub segments of the original problem can be calculated at the same time. If the total number of workers of the computer is less than the number of segments after segmentation, the segments with the number of workers are calculated first. When a worker completes the calculation first, the remaining segments are calculated immediately. Each column of data shows the number of feasible routes calculated by each computer when solving the sub problem of generating a feasible route model, the time required for each worker operation process and the time required to generate each route.

To further highlight the efficiency of the distributed Dang method, 20 groups of test examples are given below which are solved by the mathematical solver CPLEX and the distributed Dang method. The experimental comparison results are shown in Table 3. CPLEX usually uses a branch and cut algorithm to solve integer programming problems, that is, to solve a series of linear programming problems. However, in the process of solving, CPLEX's memory management is insufficient, and it takes a lot of time to solve even small scale integer programming models. CPLEX adopts automatic adjustment to solve the problem of insufficient memory, that is, by relaxing the feasible solution, the finder identifies the infeasible solution through incongruous constraints, and then corrects the model through further implementation, but the adjusted calculation will still affect the calculation performance. The table mainly compares the time required for each of the two algorithms to solve the examples. When solving small-scale examples, using CPLEX to solve the examples saves more time than using the distributed Dang method. When the problem scale is slightly increased, the time advantage of the distributed Dang method in calculating the examples is highlighted. When the problem scale is increased to a certain extent, CPLEX cannot solve the examples.

Table 3. Time comparison between CPLEX and distributed Dang methods in computing different data sets.

Example	Number of aircraft	Number of flights	CPELX	Distributed Dang method
			Time (s)	Time (s)
1	5	32	10	53
2	5	32	8	41
3	5	32	13	36
4	5	32	17	25
5	5	32	21	36
6	10	64	83	146
7	10	64	121	166
8	10	64	96	180
9	10	64	106	157
10	10	64	68	177
11	20	137	769	414
12	20	137	613	422
13	20	137	985	434
14	20	137	557	441
15	20	137	761	336
16	45	226	—	4599
17	45	226	—	5764
18	45	226	—	3846
19	45	226	—	9461
20	45	226	—	5964

According to the analysis of the data results in Table 3, when the aircraft scale is less than 10, CPLEX can finish the calculation in a relatively short time. When the aircraft scale becomes 20, the solution efficiency of the distributed Dang method is higher than CPLEX. When the aircraft scale continues to increase to 45, CPLEX cannot get the solution of the example in a reasonable time, and the distributed Dang method can still effectively solve the problem. As CPLEX is an accurate algorithm, its solution to small-scale problems can be easily obtained, but it is often powerless in the face of large-scale problems. By contrast, the distributed Dang method has more significant advantages in solving large-scale problems and, thus, fully reflects the applicability and effectiveness of distributed computing for solving such problems.

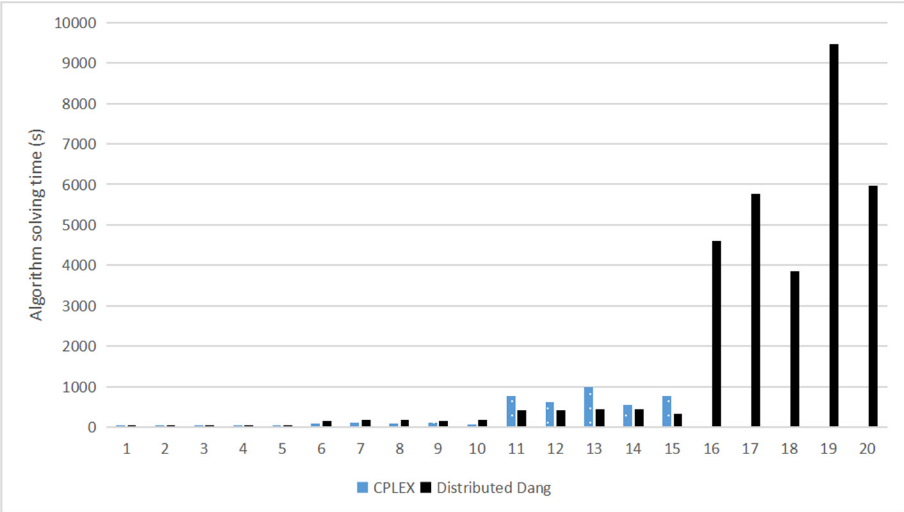

Figure 5. Comparison chart of solution times.

Table 4. Flight schedule after aircraft grounding and resumption.

Aircraft	Flight	Origin	Destination	Departure	Arrival	Delay Cost	Cancellation Cost
A1	11	CKG	NKG	11:05	13:00	0	—
	12	NKG	INC	13:50	16:10	0	—
A2	21	TAO	ICN	13:45	14:50	0	—
	22	ICN	TAO	15:40	17:00	0	—
A3	31	ZGC	JGN	14:40	15:50	0	—
A4	41	TAO	FUK	11:05	12:40	0	—
	42	FUK	TAO	13:40	16:00	0	—
A5	51	KWE	KHN	11:45	13:10	0	—
	52	KHN	NKG	13:55	15:00	0	—
A6	61	NKG	BAV	13:00	15:00	0	—
	62	BAV	NKG	15:50	17:40	0	—
A7	71	HGH	CAN	14:05	16:10	0	—
A9	91	HRB	TAO	12:00	13:55	0	—
	92	TAO	NKG	14:45	15:55	0	—
A10	101	LZO	PEK	10:00	11:10	0	—
	81	PEK	NKG	11:50	13:45	1950	—
	82	NKG	CSX	14:35	16:00	1950	—
	83	CSX	PEK	16:50	18:50	1950	—
A11	111	PVG	DAT	15:45	18:35	0	—
Cancel	102	PEK	LZO	—	—	—	10,970

All feasible route sets are represented by 0 and 1 and are solved by the distributed computing network combined with Dang algorithm. They are transformed into actual feasible routes by the heuristic conversion algorithm as the input of the aircraft reassignment model. The solution of how to reasonably allocate available aircraft to minimize the economic loss of airlines is calculated by optimizers' concert technology. The rescheduled flight schedule is shown in Table 4.

From the new flight schedule formed after reassigning flights to available aircraft, the economic loss caused by the new flight plan to the airline is 1950 + 1950 + 1950 + 10970 = 16,820 yuan. Compared with the 30,551 yuan loss caused by directly canceling all flights of aircraft A8, the new plan reduces the economic loss of the airline by about 44.9%, The results show that the feasible route generation model and aircraft reassignment model established in this paper are effective for solving the flight execution scheme with minimum loss. In order to recover the flight plan of this example, two recovery measures, delayed flight and canceled flight, were taken, that is, the departure time of flight 81 was delayed to 11:50, flights 82 and 83 were postponed, flight 102 was canceled, the flight plan of aircraft A10 was changed to 101, 81, 82, 83, and other aircraft flew according to the original plan.

Table 5. Flight schedule after the airport is closed and restored.

Aircraft	Flight	Origin	Destination	Departure	Arrival	Delay cost	Cancellation cost
A1	11	CKG	NKG	14:15	16:10	5700	—
A2	21	TAO	ICN	13:45	14:50	0	—
	22	ICN	TAO	15:40	17:00	0	—
A3	31	ZGC	JGN	14:40	15:50	0	—
A4	41	TAO	FUK	11:05	12:40	0	—
	42	FUK	TAO	13:40	16:00	0	—
A5	51	KWE	KHN	11:45	13:10	0	—
	52	KHN	NKG	14:00	14:50	150	—
	12	NKG	INC	15:30	17:50	3000	—
A7	71	HGH	CAN	14:05	16:10	0	—
A8	81	PEK	NKG	14:00	15:55	5850	—
A9	91	HRB	TAO	12:00	13:55	0	—
	92	TAO	NKG	14:45	15:55	0	—
A10	101	LZO	PEK	10:00	11:10	0	—
	102	PEK	LZO	11:55	15:30	0	—
A11	111	PVG	DAT	15:45	18:35	0	—
Cancel	61	NKG	BAV	—	—	—	12,524
	62	BAV	NKG	—	—	—	9352
	82	NKG	CSX	—	—	—	8465
	83	CSX	PEK	—	—	—	11,823

The problem of flight interruption caused by aircraft failure has been improved after being processed by the recovery model. Next, we will verify the problem of interruption caused by airport closure. The airport NKG was closed from 12:00 to 14:00 due to thunderstorm weather. If no recovery measures are taken for the flight plan, the affected flights will be canceled directly, which will result in an economic loss of 80,252 yuan. Table 5 shows the aircraft operation plan for reassigning flight tasks due to airport closure. Delay flights 11, 52 and 81; assign flight 12 to aircraft A5 for execution; and cancel flights 61, 62, 82 and 83. Two measures, flight delay and flight cancellation, were taken in the plan after recovery, resulting in a total loss of 56,864 yuan, saving 23,388 yuan compared with the flight plan before recovery. Through Table 6, we can more intuitively observe the loss difference before and after the flight plan recovery in the case of interruption.

Table 6. Comparison of recovery results.

Fault type	Number of canceled flights		D-value	Loss		D-value
	No action taken	Take steps		No action taken	Take steps	
Aircraft grounded	3	1	2	30,551	16,820	13,737
Airport closure	7	4	3	80,252	56,864	23,388

5. Conclusions

To reduce the loss caused by flight interruption to restore the aircraft route to realize intelligent unmanned aviation, this paper proposes a distributed integer programming calculation method to restore the aircraft route. The problem of restoring the route is divided into two sub problems for modeling and solving. The first sub problem is modeled as a feasible route generation model, and an average segmentation method is proposed to divide the problem space into independent segments, which is convenient for simultaneous calculation using the distributed computing network. The example shows that the distributed Dang method is more efficient than CPLEX in solving the feasible route, and the feasible routes generated by the dictionary order are closer to the original plan than the plan recovered by the common method, with less disturbance impact. The second sub problem is modeled as the aircraft reassignment model, and the result of reassigning the aircraft is solved using the feasible route calculated in the previous problem to obtain the final restored flight plan. The examples of flight interruption caused by aircraft failure and airport closure are considered. Compared with the decision of directly canceling the affected flights, the flight plan after resuming scheduling greatly reduces the economic losses to airlines. The flight plan after resuming scheduling has reduced the number of flight cancellations by 2 and 3, respectively, compared to the decision to directly cancel the affected flights, and has reduced economic losses by 44.96% and 29.14%, respectively. It can also be applied in unmanned air transportation systems in the future.

The feasible route generative model and aircraft reassignment model proposed in this paper can quickly respond to emergencies and develop new aircraft travel plans when handling flight interruption events. This proves the contribution of this study to energy conservation and cost savings.

The future work plan is to continue to expand the research of distributed computing. If the segmentation method proposed in this paper is more complex, it requires a large number of computing processors to complete the calculation together to ensure that the calculation results can be obtained in a short time. In addition, the study in this paper only considers the plan of resuming flight operation, and we can continue to study the plan of passenger boarding and crew scheduling in the future.

Acknowledgments

This research was funded by the NSFC under grant nos. 61803279, in part by the Postgraduate Research & Practice Innovation program of Jiangsu Province under Grant no. KYCX22_3277 and SJCX22_1585, in part by the Qing Lan Project of Jiangsu, in part by the China Postdoctoral Science

Foundation under Grant no. 2020M671596 and 2021M692369, in part by the Suzhou Science and Technology Development Plan Project (Key Industry Technology Innovation) under Grant no. SYG202114, in part by the Open Project Funding from Anhui Province Key Laboratory of Intelligent Building and Building Energy Saving, Anhui Jianzhu University, under Grant no. IBES2021KF08.

Conflict of interest

The authors declare there is no conflict of interest.

References

1. J. Clausen, A. Larsen, J. Larsen, N. J. Rezanova, Disruption management in the airline industry—Concepts, models and methods, *Comput. Oper. Res.*, **37** (2010), 809–821. https://doi.org/10.1016/j.cor.2009.03.027
2. B. Jiang, Z. Wu, H. R. Karimi, A distributed dynamic event-triggered mechanism to HMM-based observer design for H∞ sliding mode control of Markov jump systems, *Automatic*, **142** (2022), 110357. https://doi.org/10.1016/j.automatica.2022.110357
3. T. K. Liu, C. R. Jeng, Y. H. Chang, Disruption management of an inequality-based multi-fleet airline schedule by a multi-objective genetic algorithm, *Transp. Plann. Technol*., **31** (2008), 613–639. https://doi.org/10.1080/03081060802492652
4. Z. Wu, B. Jiang, H. R. Karimi, A logarithmic descent direction algorithm for the quadratic knapsack problem, *Appl. Math. Comput.*, **369** (2020), 124854. https://doi.org/10.1016/j.amc.2019.124854
5. D. Teodorović, S. Guberinić, Optimal dispatching strategy on an airline network after a schedule perturbation, *Eur. J. Oper. Res.*, **15**(1984), 178–182. https://doi.org/10.1016/0377-2217(84)90207-8
6. A. I. Jarrah, G. Yu, N. Krishnamurthy, A. Rakshit, A decision support framework for airline flight cancellations and delays, *Transp. Sci.*, **27** (1993), 266–280. https://doi.org/10.1287/trsc.27.3.266
7. J. M. Cao, A. Kanafani, Real-time decision support for integration of airline flight cancellations and delays part I: mathematical formulation, *Transp. Plann. Technol.*, **20** (1997), 183–199. https://doi.org/10.1080/03081069708717588
8. T. Zhou, J. Lu, W. Zhang, P. He, B. Niu, Irregular flight timetable recovery under covid-19: An approach based on genetic algorithm, in *Data Mining and Big Data: 6th International Conference*, (2021), 240–249. https://doi.org/10.1007/978-981-16-7476-1_22
9. S. Wang, F. Xu, W. Yang, Z. Ma, Application of greedy random adaptive search algorithm (GRASP) in flight recovery problem, *Int. J. Adv. Network Monit. Controls*, **3** (2018), 39–44. https://doi.org/10.21307/ijanmc-2018-008
10. H. Lin, Z. Wang, Fast variable neighborhood search for flight rescheduling after airport closure, *IEEE Access*, **6** (2018), 50901–50909. https://doi.org/10.1109/ACCESS.2018.2869842
11. N. Kenan, A. Jebali, A. Diabat, The integrated aircraft routing problem with optional flights and delay considerations, *Transp. Res. Part E*, **118** (2018), 355–375. https://doi.org/10.1016/j.tre.2018.08.002
12. H. Lin, Z. Wang, Flight scheduling for airport closure based on sequential decision, in *2018 4th International Conference on Information Management (ICIM)*, (2018), 241–245. https://doi.org/10.1109/INFOMAN.2018.8392843

13. D. Teodorović, G. Stojković, Model for operational daily airline scheduling, *Transp. Plann. Technol.*, **14** (1990), 273–285. https://doi.org/10.1080/03081069008717431
14. D. Teodorović, G. Stojković, Model to reduce airline schedule disturbances, *J. Transp. Eng.*, **121** (1995), 324–331. https://doi.org/10.1061/(ASCE)0733-947X(1995)121:4(324)
15. Z. Wu, H. R. Karimi, C. Dang, An approximation algorithm for graph partitioning via deterministic annealing neural network, *Neural Networks*, **117** (2019), 191–200. https://doi.org/10.1016/j.neunet.2019.05.010
16. Z. Wu, Q. Gao, B. Jiang, H. R. Karimi, Solving the production transportation problem via a deterministic annealing neural network method, *Appl. Math. Comput.*, **411** (2021), 126518. https://doi.org/10.1016/j.amc.2021.126518
17. Z. Wu, H. R. Karimi, C. Dang, A deterministic annealing neural network algorithm for the minimum concave cost transportation problem, *IEEE Trans. Neural Networks Learn. Syst.*, **31** (2019), 4354–4366. https://doi.org/10.1109/TNNLS.2019.2955137
18. L. B. Fogaça, E. Henriqson, G. C. Junior, F. Lando, Airline disruption management: A naturalistic decision-making perspective in an operational control centre, *J. Cognit. Eng. Decis. Making*, **16** (2022), 3–28. https://doi.org/10.1177/15553434211061024
19. J. Lee, L. Marla, P. Vaishnav, The impact of climate change on the recoverability of airline networks, *Transp. Res. Part D*, **95** (2021), 102801. https://doi.org/10.1016/j.trd.2021.102801
20. S. Bouarfa, J. Müller, H. Blom, Evaluation of a multi-agent system approach to airline disruption management, *J. Air Transp. Manage.*, **71** (2018), 108–118. https://doi.org/10.1016/j.jairtraman.2018.05.009
21. M. F. Argüello, J. F. Bard, G. Yu, A GRASP for aircraft routing in response to groundings and delays, *J. Comb. Optim.*, **1** (1997), 211–228. https://doi.org/10.1023/A:1009772208981
22. Z. Wu, B. Li, C. Dang, F. Hu, Q. Zhu, B. Fu, Solving long haul airline disruption problem caused by groundings using a distributed fixed-point computational approach to integer programming, *Neurocomputing*, **269** (2017), 232–255. https://doi.org/10.1016/j.neucom.2017.02.091
23. C. Dang, An increasing-mapping approach to integer programming based on lexicographic ordering and linear programming, in *The Ninth International Symposium on Operations Research and Its Applications. Lecture Notes in Operations Research*, **12** (2010), 55–60. https://doi.org/10.1201/b17196-11
24. C. Dang, Y. Ye, A fixed point iterative approach to integer programming and its distributed computation, *Fixed Point Theory Appl.*, **1** (2015), 1–15. https://doi.org/10.1186/s13663-015-0429-8