Received August 1, 2017, accepted August 20, 2017, date of publication August 30, 2017, date of current version October 12, 2017.

Digital Object Identifier 10.1109/ACCESS.2017.2747155

Solving Multiple Fleet Airline Disruption Problems Using a Distributed-Computation Approach to Integer Programming

ZHENGTIAN WU[1,2], BENCHI LI[2], AND CHUANGYIN DANG[2]
[1]School of Electronic and Information Engineering, Suzhou University of Science and Technology, Suzhou 215009, China
[2]Department of Systems Engineering and Engineering Management, City University of Hong Kong, Hong Kong

Corresponding author: Zhengtian Wu (wzht8@mail.usts.edu.cn)

This work was supported in part by the GRF:CityU 11301014 of Hong Kong SAR Government, in part by the National Nature Science Foundation of China under Grant 71471091, Grant 71271119, Grant 61672371, Grant 51375323, and Grant 61472267, in part by the Natural science fund for colleges and universities in Jiangsu Province under Grant 17KJD110008, and in part by the Social Development Project of Jiangsu Provincial under Grant BE2017663.

ABSTRACT The execution of the airline operation is often deviated from the original schedule due to some unexpected disruptions, such as aircraft breakdowns and severe weather conditions. In this situation, a recovery plan is needed to get the irregular operation back to normal to minimize the losses of the airline. To produce recovery plans and solve the airline disruption problems, a novel modified traveling salesman problem model is proposed to generate sets of the feasible flight routes for each aircraft fleet type. Then, the feasible flight routes are reassigned to the available aircrafts in each fleet to form a recovery plan. Numerical results show that the approach proposed in this paper is efficient and promising.

INDEX TERMS Airline disruption management, irregular operation, integer programming, distributed computation, OpenMP.

I. INTRODUCTION

Generally, traveling by aircrafts is the most convenient and time-saving way for long distance transportation. Therefore, the punctuality of airlines is significant to the airline carriers because unpunctuality always causes great inconvenience to the passengers and brings losses to airlines. To pursue a good punctuality, the operation of an airline requires an elaborate construction of an airline schedule planning, which includes flight scheduling, fleet assignment, aircraft routing and crew scheduling. The ideal execution of the airline operation is identical to the flight schedules [1], but it is seldom operated as original plan due to unexpected disruptions such as aircraft breakdowns, crew absences, severe weather conditions and air traffic & airport restrictions. Thus aircrafts, crews and passengers are affected by the disruptions. The disruptions can cause misconnect, rest and duty problems to the schedules of the crews [2]. The passengers may miss their connecting flights or other transportation, which leads to economy losses. For the airlines, the ground time they spend at airports between two consecutive flights is minimized to maximize the utilization of each aircraft. Therefore, any disruption can cause downstream impact on the subsequent flights of the disrupted aircrafts since the aircrafts and crews may not be available at the scheduled time of the downstream flights [2], [3]. The consequential disruption of the downstream flights is called delay propagation, and it affects the airline operations much more than the initial delay. The disruptions not only cause great inconvenience to the passengers but also bring losses to airlines. For a typical airline, losses caused by the disruptions approximately take up 10% of its scheduled revenue according to the paper [4].

In mainland China, the punctuality performance of all the airlines in 2010 is reported in [5]. There are 1,888 thousands scheduled flights in the major airline companies. 1,431 thousands flights perform as scheduled, but 457 thousands flights or 24.2% of the total flights perform irregularly. For the middle and small airlines, of the total 260 thousands scheduled flights, there are 81 thousands flights operated irregularly. The percentage is up to 31.2%. The total losses caused by the irregularities reaches CNY2.1 billion in 2002, and it is estimated to increase to CNY7.6 billion in 2020 [6]. The general manager from China Eastern Air Holding Co. points out that the direct operating cost of China Eastern Airlines is CNY1,000 for each minute's delay in 2011, and the cost does not cover the compensation and the service to the passengers [7].

In the Europe, it is revealed by European Organization for the Safety of Air Navigation (EUROCONTROL) [8] and *Central Office for Delay Analysis (CODA)*[1] [9] that 18% of the flights are delayed on arrival by more than 15 minutes in 2011.EUROCONTROL [8] shows the estimated cost of ATFM delays decreases from C2.2 billion in 2010 to C1.45 billion in 2011, but it is still higher than that in 2009 by 21%. In the United States, the average percentage of the punctual flights from September 1987 to December 2003 is 78.9% [3]. The lowest point is only 72.6% that happened in December 2000 [3]. The U.S. reports that the direct aircraft operating cost per minute is $65.19 in 2010, and it increases by 6% versus 2009 [10]. The total delay costs come up to $6,475 millions in 2010.

Since the disruptions can result in huge revenue losses to airlines inevitably, the punctuality performance affects the profit of the airlines. However, the research in [11] shows that most airlines with high punctuality rates appear to be more profitable than those low punctuality rates. When the airline disruptions happen, producing a recovery plan can reduce the losses and increase the punctuality.

It is a complex task to produce recovery plans since many resources such as crews, aircrafts and passengers have to be reassigned and this problem have attracted researchers' interests since 1980s. A branch and bound method is used to produce the recovery plan to minimize total passenger delays [12], and the work is further extended in the papers [13], [14]. All of them model the airline disruption problem as a connection network which is also adopted in [15] and [16]. The papers [17], [18] use the time-line network to formulate the problem. Other papers [19], [20] propose a time-band network to model the airline disruption problem. A framework for the integrated recovery that considers aircraft, crew and passenger together is proposed in [21]. In terms of solution methods, the research in [18] and [19] apply heuristics to solve the airline disruption problems based on the models they formulated, while the majority of the other researchers mentioned before use integer programming to solve the problems. An introduction to airline disruption management is given [22]. An inequality-based multi-objective genetic algorithm (MMGA) that is capable of solving multiple objective airline disruption problems is developed [23]. For more theoretical descriptions and comparisons regarding the airline disruption management, one may refer to the review by the paper [1]. Some new method and theory [24]–[27] could also be used to this airline disruption problems.

The multiple fleet airline disruption problems consist of two subproblems. In the first subproblem, sets of the feasible flight routes for each fleet are generated by Dang and Ye's method [28] in a distributed computation network. In the second subproblem, the feasible flight routes are reassigned to the available aircrafts in each fleet to form a recovery plan. The first subproblem is mainly focused in this paper and a modified Traveling Salesman Problem (TSP) model is proposed to formulate the first subproblem. The flight legs are matched to form flight pairs which are the decision variables in the modified TSP model, and this model is different from any model that is formulated for planning or recovery purposes in literature. The solution of the problem is a sequence of flight pairs, and any of the two consecutive flight pairs are joint in the same flight legs. So the sequence of flight A distributed computation is proposed based on Dang and Ye's iterative method for integer programming. Using this method, the feasible flight routes for an aircraft are obtained sequentially in the lexicographical order from the original flight route of the aircraft. The feasible flight routes for all the aircrafts in all the fleet types can be generated simultaneously in the distributed computation. Not all the feasible flight routes are needed for the second subproblem. Only by providing partial feasible flight routes can a solution of the second subproblem be found. When sufficient feasible flight routes are generated, solutions that are better than the solutions in literature can be found from the computational experience in Section IV.

[1] CODA is a service of EUROCONTROL

The rest of this paper is organized as followed. Section II gives the mathematical formulations of the two subproblems. Dang and Ye's method and the distributed computation are introduced in Section III. In Section IV, performance comparisons are presented. Finally, a conclusion of this paper is given in Section V.

II. PROBLEM FORMULATION

The problem that is focused on this paper is to reassign the available aircrafts during a disruption period. The disruption is caused by severe weather or aircraft mechanical malfunction or any other reasons and results in that some aircrafts are unavailable for a certain amount of time in their original schedule or even the whole day. All the flights served by the unavailable aircrafts may be canceled if no recovery action is taken. The cancelations can cost the air carriers a lot and dissatisfy the passengers. But through reassigning the flight legs of the unavailable aircrafts to the remaining available aircrafts, a recovery schedule of the airline with a minimum impact caused by the disruption can be found.

Normally most air carriers have multiple fleet types, and they have different characteristics. Of all the characteristics, the passenger capacity is the most important one to this research. During the recovery process of the disrupted airline schedule, an affected flight leg can only be reassigned to an aircraft with a passenger capacity that is larger than the actual passenger load of the flight leg.

The existence of the aircraft substitutions in the multiple fleet airline disruption problem makes it more difficult than the single fleet problem since the sets of the flight legs that can be assigned to each fleet are different. To solve this problem, a distributed implementation of the iterative method [28] for integer programming is proposed to generate feasible flight routes for each fleet. Then the generated feasible flight routes are used to construct an aircraft reassignment to recovery

TABLE 1. Sample schedule.

Aircraft	Flight ID	Origin	Destination	Departure time	Arrival time	Duration	Cancelation cost
A	11	BOI	SEA	1410	1520	1:10	7350
	12	SEA	GEG	1605	1700	0:55	10231
	13	GEG	SEA	1740	1840	1:00	7434
	14	SEA	BOI	1920	2035	1:15	14191
B	21	SEA	BOI	1545	1700	1:15	11189
	22	BOI	SEA	1740	1850	1:10	12985
	23	SEA	GEG	1930	2030	1:00	11491
	24	GEG	SEA	2115	2215	1:00	9581
C	31	GEG	PDX	1515	1620	1:05	9996
	32	PDX	GEG	1730	1830	1:00	15180
	33	GEG	PDX	1910	2020	1:10	17375
	34	PDX	GEG	2100	2155	0:55	15624

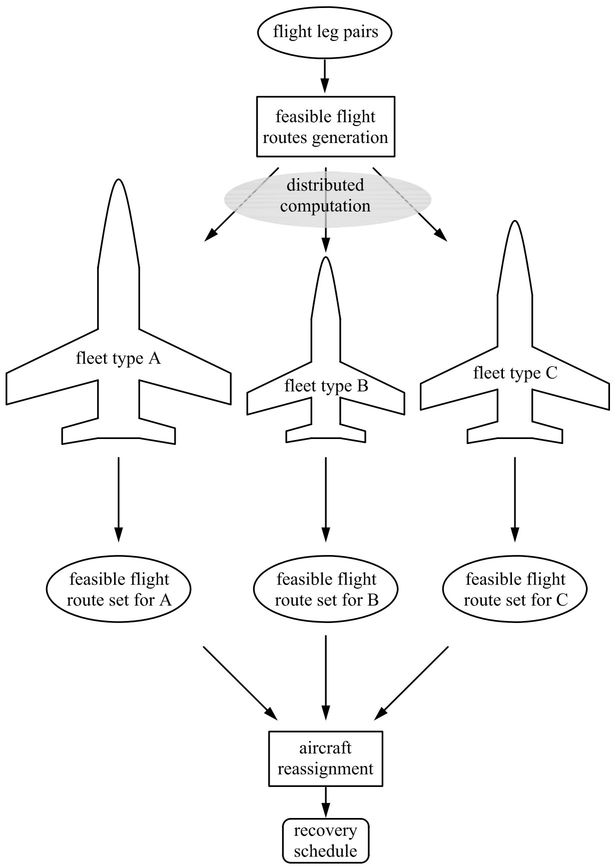


FIGURE 1. Process of the two subproblems.

the disrupted schedule. To find an aircrafts reassignment that minimizes the loss of the disruption and the deviation from the original schedule, the process concerns two subproblems as illustrated in Figure 1. First, sets of the feasible flight routes for each fleet are generated simultaneously in the first subproblem formulated in Subsection II-A. Then for the aircrafts in each fleet, feasible flight routes are chosen from the feasible flight route set that corresponds to the fleet to form a recovery schedule in the second subproblem given in Subsection II-B.

A. FEASIBLE FLIGHT ROUTES GENERATION

The decision variables in the feasible flight routes generation model of our previous works [29]–[31] are flight legs, and the sequence of the flight legs cannot be decided in the previous models. Thus when the numerical solutions of the model are transformed into the actual flight routes, plenty of the actual flight routes are useless because there exists very long delay or grounding time between two flight legs. Thus enormous feasible solutions are needed to generate the actual feasible flight routes. To improve the efficiency, a modified TSP model is proposed to generate feasible flight routes in this paper. The original flight legs are matched to form flight pairs which are the decision variables in the modified TSP model. The flight pairs are generated under the constraint that the destination of the first flight leg is the same as the origin of the second flight leg. A sample schedule [32] in Table 1 is used to illustrate the generation of the flight pairs.

The TSP requires that all the cities have to be visited exactly once, and the visit returns to the beginning city to form a tour. However, all the flight legs are chosen to form a route at most once in the feasible flight routes generation problem. So some of the flight legs may not be chosen. Two dummy stations named Source and Sink are introduced to indicate the first flight leg and the last flight leg of a flight route respectively. The Source combines with the original station of each flight route and the Sink combines with the terminal station of each flight route in the original schedule to form dummy flight legs. The dummy flight legs match with the flight legs in the original schedule to form dummy flight pairs as in Table 2. Each resulting feasible flight pair route must originates from one of the source flight pairs and terminates at one of the sink flight pairs.

Table 3 displays the flight pairs and the dummy flight pairs that are constructed by the flight legs and the dummy flight legs. The symbol $\surd$ in Table 3 is a valid flight pair which consists of two consecutive flight legs while the symbol $\times$ is an invalid flight pair. The flight legs in the second column are the first flight legs in the flight pairs, and the flight legs in the second row are the second flight legs in the flight pairs. If an aircraft have flew the first flight leg in a flight pair, it must have to fly the second flight leg in the flight pair. Some of the second flight legs in the flight pairs are delayed a long period of time such as the flight pair that consists of flight leg 34 and flight leg 13, and some of the turnaround time between the two flight legs in the flight pairs are too long such as the flight pair that consists of flight leg 11 and flight leg 23. They can

TABLE 2. Dummy flight pairs.

Source flight pair		Sink flight pair	
First	Second	First	Second
Source→B ID:0	B→S 11	S→B 14	B→Sink ID:3
	B→S 22	S→B 21	
Source→S ID:1	S→G 12	B→S 11	S→Sink ID:4
	S→B 14	G→S 13	
	S→B 21	B→S 22	
	S→G 23	G→S 24	
Source→G ID:2	G→S 13	S→G 12	G→Sink ID:5
	G→S 24	S→G 23	
	G→P 31	P→G 32	
	G→P 33	P→G 34	

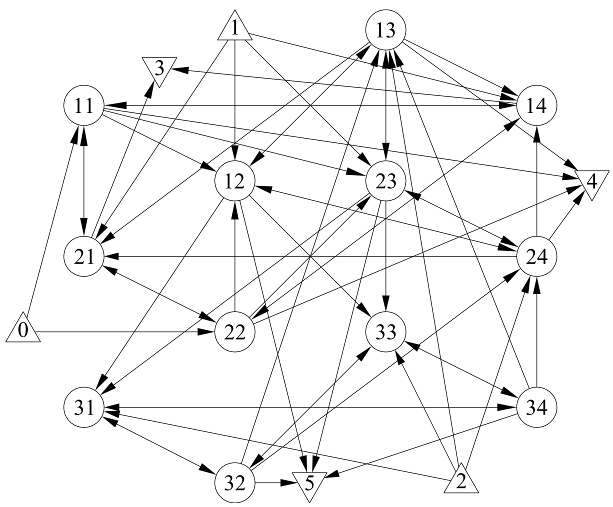


FIGURE 2. Network of the feasible flight routes generation problem.

be marked × if the maximum delay time and the maximum turnaround time are given.

Figure 2 is the network of the feasible flight routes generation problem. The arcs represent the valid flight pairs, the valid dummy source flight pairs and the valid dummy sink flight pairs which are symbolized $\surd$ in Table 3. For the arcs with a single arrow, only one flying direction is allowed. For the arcs with double arrows, both of the two flying directions are allowed and they represent two valid flight pairs. The circular nodes represent the flight legs, and the triangular nodes and the inverse triangular nodes respectively represent the dummy source flight legs and the dummy sink flight legs. The problem is to find a set of feasible flight route for an involved fleet type, and each feasible flight route is a combination of the arcs in Figure 2. Only partial arcs and nodes are chosen to form the combination, and they must satisfy the following conditions:

- Only nodes with actual passenger loads that do not exceed the aircraft capacity of the involved fleet can be chosen,
- Only one of the arcs that leave the triangular nodes is chosen,
- Only one of the arcs that enter the inverse triangular nodes is chosen,
- Each circular node in the chosen arcs is entered and left exactly once,
- No subtour exists.

Once all the requirements are met, the resulting chosen arcs are connected to each other in a sequence that originates from one of the triangular nodes and terminates at one of the inverse triangular nodes. The actual passenger loads of each node are within the aircraft capacity of the fleet type, so the feasible flight routes that are obtained from the chosen arcs can be flew by the aircrafts in the fleet type.

To construct the mathematical formulation of the network in Figure 2, $F = \{1, \cdots, i, \cdots, j, \cdots, n\}$ is defined to denote the set of circular nodes that represent the flight legs, $\mathscr{A}$ is introduced to denote the set of arcs that represent the valid flight pairs ($(i, j) \in \mathscr{A}$). Decision variables x_{ij} are introduced to formulate the feasible generation problem, and they denote the valid flight pairs. $x_{ij} = 1$ if the flight leg j immediately follows the flight leg i in the flight route, $x_{ij} = 0$ otherwise. For the invalid flight pairs ($(i, j) \notin \mathscr{A}$) which are symbolized × in Table 3, $x_{ij} = 0$ always holds.

Let $F_{source} = \{1, \cdots, u, \cdots, l\}$ denotes the set of triangular nodes that represent the dummy source flight legs and $\mathscr{A}_{source}$ denotes the set of arcs that represent the valid dummy source flight pairs ($(u, i) \in \mathscr{A}_{source}$), and $F_{sink} = \{1, \cdots, v, \cdots, m\}$ denotes the set of inverse triangular nodes that represent the dummy sink flight legs and $\mathscr{A}_{sink}$ denotes the set of arcs that represent the valid dummy sink flight pairs ($(i, v) \in \mathscr{A}_{sink}$). Decision variables s_{ui} and k_{iv} denote the valid dummy source flight pairs and the valid dummy sink flight pairs respectively. $s_{ui} = 1$ if the first flight leg of a flight route is i, $s_{ui} = 0$ otherwise. $k_{iv} = 1$ if the last flight leg of a flight route is i, $k_{iv} = 0$ otherwise. For the invalid dummy source flight pairs ($(u, i) \notin \mathscr{A}_{source}$) and the invalid dummy sink flight pairs ($(i, v) \notin \mathscr{A}_{sink}$) which are symbolized × in Table 3, $s_{ui} = 0$ and $k_{iv} = 0$ always hold.

The notations used in the formulation are given below.

Indices

i, j, w	flight leg indices
u	dummy source flight index
v	dummy sink flight index

Sets

F	set of the flight legs
F_{source}	set of the dummy source flight legs
F_{sink}	set of the dummy sink flight legs
$\mathscr{A}$	set of the flight pairs
$\mathscr{A}_{source}$	set of the dummy source flight pairs
$\mathscr{A}_{sink}$	set of the dummy sink flight pairs

TABLE 3. Full flight pair matrix.

		Flight ID												Dummy sink flight ID		
		11	12	13	14	21	22	23	24	31	32	33	34	3	4	5
Flight ID	11	×	√	×	√	√	×	√	×	×	×	×	×	×	√	×
	12	×	×	√	×	×	×	×	√	√	×	√	×	×	×	√
	13	×	√	×	√	√	×	√	×	×	×	×	×	×	√	×
	14	√	×	×	×	×	√	×	×	×	×	×	×	√	×	×
	21	√	×	×	×	×	√	×	×	×	×	×	×	√	×	×
	22	×	√	×	√	√	×	√	×	×	×	×	×	×	√	×
	23	×	×	√	×	×	×	×	√	√	×	√	×	×	×	√
	24	×	√	×	√	√	×	√	×	×	×	×	×	×	√	×
	31	×	×	×	×	×	×	×	×	×	√	×	√	×	×	×
	32	×	×	√	×	×	×	×	√	√	×	√	×	×	×	√
	33	×	×	×	×	×	×	×	×	×	√	×	√	×	×	×
	34	×	×	√	×	×	×	×	√	√	×	√	×	×	×	√
Dummy source flight ID	0	√	×	×	×	×	√	×	×	×	×	×	×	×	×	×
	1	×	√	×	√	√	×	√	×	×	×	×	×	×	×	×
	2	×	×	√	×	×	×	×	√	√	×	√	×	×	×	×

Coefficients

- f_{min} minimum number of flight legs required by the problem
- f_{max} maximum number of flight legs required by the problem
- t_i duration of the flight leg i including the turnaround time
- T total length of time from the departure of the earliest flight to
  the departure curfew time of stations
- C aircraft capacity of the involved fleet
- c_i actual passenger loads of the flight leg i

Variables

- x_{ij} equal to 1 if the flight leg j immediately follows the flight leg i
  in the flight route; 0 otherwise
- s_{ui} equal to 1 if the first flight leg of the flight route is i; 0 otherwise
- k_{iv} equal to 1 if the last flight leg of the flight route is i; 0 otherwise

Feasible Route Generation Mathematical Formulation

$$\exists x_{ij}, s_{ui}, k_{iv} \in \{0, 1\},$$
$$\forall (i,j) \in \mathscr{A}, (u,i) \in \mathscr{A}_{source}, (i,v) \in \mathscr{A}_{sink} \tag{1a}$$

subject to (flight number constraint)

$$f_{min} - 1 \leq \sum_{(i,j)\in\mathscr{A}} x_{ij} \leq f_{max} - 1 \tag{1b}$$

(flight time constraint)

$$\sum_{(i,j)\in\mathscr{A}} t_j x_{ij} + \sum_{(u,i)\in\mathscr{A}_{source}} t_i s_{ui} \leq T, \tag{1c}$$

(sink node conservation)

$$\sum_{(i:(i,j)\in\mathscr{A})} x_{ij} - k_{jv} \geq 0, \quad \forall (j,v) \in \mathscr{A}_{sink}, \tag{1d}$$

(source node conservation)

$$sum_{(j:(i,j)\in\mathscr{A})} x_{ij} - s_{ui} \geq 0, \quad \forall (u,i) \in \mathscr{A}_{source}, \tag{1e}$$

(flow-out node conservation)

$$\sum_{(w:(j,w)\in\mathscr{A})} x_{jw} + \sum_{(v:(j,v)\in\mathscr{A}_{sink})} k_{jv} - x_{ij} \geq 0,$$
$$\forall (i,j) \in \mathscr{A}, \tag{1f}$$

(flow-in node conservation)

$$\sum_{(w:(w,i)\in\mathscr{A})} x_{wi} + \sum_{(u:(u,i)\in\mathscr{A}_{source})} s_{ui} - x_{ij} \geq 0,$$
$$\forall (i,j) \in \mathscr{A}, \tag{1g}$$

(at most one flow-in node)

$$\sum_{(j:(i,j)\in\mathscr{A})} x_{ij} + \sum_{(v:(i,v)\in\mathscr{A}_{sink})} k_{iv} \leq 1, \quad \forall i \in F, \tag{1h}$$

(at most one flow-out node)

$$\sum_{(i:(i,j)\in\mathscr{A})} x_{ij} + \sum_{(u:(u,j)\in\mathscr{A}_{source})} s_{uj} \leq 1,$$
$$\forall j \in F, \tag{1i}$$

(source node cover)

$$\sum_{(u,i)\in\mathscr{A}_{source}} s_{ui} = 1, \tag{1j}$$

(sink node cover)

$$\sum_{(i,v)\in\mathscr{A}_{sink}} k_{iv} = 1, \tag{1k}$$

(subtour elimination)

$$\sum_{((i,j)\in\mathscr{A}:i\in U, j\in U)} x_{ij} \leq |U| - 1,$$
$$\forall U \subset F \text{ with } 2 \leq |U| \leq |F| - 2, \tag{1l}$$

(capacity constraint)

$$\sum_{((i,j)\in\mathscr{A}:c_i>C||c_j>C)} x_{ij} = 0. \tag{1m}$$

Prior to the construction of this model, the sequence of the flight pairs are rearranged according to the position of the disrupted flight legs. If some earlier flight legs in a flight route are disrupted, the flight pairs are rearranged in the decreasing order of the departure time of the first flight leg. If the departure times of two flight pairs' first flight legs are the same, the departure times of the second flight legs are compared. For an original flight route, new feasible flight routes with different early flight legs can be found using Dang and Ye's method. If the disrupted flight legs appear in the back of a flight route, the flight pairs are rearranged in the increasing order of the departure time of the first flight leg. Dang and Ye's method can compute new feasible flight routes with different back flight legs when an original flight route is given. If the whole flight route is disrupted for a large problem, both the increasing order of the departure time sequence and the decreasing order of the departure time sequence participate in the computation respectively.

The model aims at enumerating all the feasible flight pair routes under the constraints (1b) to (1l). The constraint (1b) limits the number of the flight legs to the requirement given by the problem. The constraint (1c) is a knapsack problem constraint, and it ensures that the total durations of each chosen flight leg do not exceed the total length of time from the departure of the earliest flight leg to the departure curfew time of stations. The constraint (1d) and the constraint (1e) ensure that if a dummy sink or source flight pair is chosen, there must be a flight pair connecting it. The constraint (1f) ensures that if a flight pair is chosen, there must be a flight pair or a dummy sink flight pair connecting it on the second flight leg of the chosen flight pair. The constraint (1g) ensures that if a flight pair is chosen, there must be a flight pair or a dummy source flight pair connecting it on the first flight leg of the chosen flight pair. The constraint (1h) and the constraint (1i) restrict that each flight leg is flew at most once. The constraint (1j) and the constraint (1k) impose that only one source node and one sink node exist. The constraint (1l) [33] is used to eliminate subtours. The constraint (1m) excludes flight pairs that consist of flight legs with actual passenger loads exceeding the aircraft capacity of the involved fleet. For the dummy flight pairs that consist of flight legs with actual passenger loads exceeding the aircraft capacity, they can be excluded under the constraints (1d), the constraint (1e) and the constraint (1m).

The constraint (1h) and the constraint (1i) are modified from the TSP constraint which ensures that each city is entered and left exactly once [33]. They meet the requirement that each flight leg is flew at most once, but they cannot ensure that each flight pair is connected by another flight pair. For the first/second flight leg in a chosen flight pair, maybe none of the flight pairs that contain it as the second/first flight leg is chosen. So the constraints (1d) to (1g) are constructed to make sure that all the chosen flight pairs connect to each other only once, resulting in that each flight leg in the chosen flight pairs is flew exactly once.

The paper [33] claims that the number of the constraint (1l) is nearly $2^{|F|}$. But the number is limited to a small amount in the airline disruption problem. Let $U_s = \{(i,j) \in \mathscr{A} : i \in U, j \in U\}$, so $|U_s| = |U|$ holds in the TSP. But $|U_s| \leq |U|$ always holds in the airline disruption problem. Considering the situation that $U = \{i, j\}$ and $|U| = 2$. If x_{ij} is a valid flight pair ($(i,j) \in \mathscr{A}$), x_{ji} can be an invalid flight pair ($(j,i) \notin \mathscr{A}$). The flight leg i's destination is the same as the flight leg j's origin, but the flight leg j's destination may not be the same as the flight leg i's origin. Besides, the change of the flight leg sequence from x_{ij} to x_{ji} can result in a huge delay time between two flight legs in x_{ji}. Once the delay time exceeds the maximum delay time given by the problem, x_{ji} should be classified as an invalid flight pairs. Both $x_{11,14}$ and $x_{14,11}$ are valid flight pairs in Table 3, but the flight leg 11 has to delay nearly 4.5 hours in the flight pair $x_{14,11}$. $x_{14,11}$ should be classified as an invalid flight pairs once a maximum delay time is given and the maximum delay time is short than the delay time in $x_{14,11}$. Since x_{ji} does not exist, $|U_s| = 1 \leq |U| - 1$ always holds, and the constraint (1l) which does not work can be deleted for this situation.

The feasible solutions of (1) are sequences of the flight pairs that begin in a dummy source flight pair and end in a dummy sink flight pair, and they can be easily transformed to sequences of the flight legs which are the feasible flight routes. Although the constraint (1c) ensures that the total durations of each chosen flight leg do not exceed the total length of available flying time, the actual total flying time of a feasible flight route is much longer than the sum of the durations of each flight leg in the feasible flight route. The reason is that there may exists big time intervals between two consecutive flight legs in the feasible flight route. Generally speaking, an aircraft can not departure earlier than the original departure time in the recovery schedule, and it have to wait in the airport during the time intervals. If the total flying time of a feasible flight route is too long that the departure time of the last flight leg exceeds the the departure curfew time of stations, the feasible flight route will be deleted.

Let P be the set of all the feasible flight routes that includes the original flight routes and S be the set of all the airports. The research [34] asserts that the $|P|$ is very large with respect to $|F|$ and $|S|$. If the length of each flight route is restricted to at most v flight legs, where $v \ll |F|$ and $v < |S|$, [34] shows that $|P|$ is bounded below by a function $\Phi(v)$, where

$$\Phi(v) = O(2^v) \tag{2}$$

Thus, $|P|$ is exponential with respect to the maximum flight route length v. No detail on generating P is given in the papers [15], [34], [35]. In this paper, the feasibility problem (1) is proposed to generate P. For small $|F|$ and $|S|$, it's not difficult and time-consuming to generate all the feasible flight routes using CPLEX CP Optimizer. However, when $|F|$ and $|S|$ are large, $|P|$ is extremely large and CPLEX CP Optimizer can't generate all the feasible flight routes in a reasonable amount of time. To tackle this difficulty, a distributed computational approach to integer programming based on Dang and Ye's method is proposed to solve the feasibility problem (1). Details of this method are given in Section III.

B. AIRCRAFTS REASSIGNMENT

Once the first subproblem (1) is solved, the feasible flight routes for one involved aircraft fleet can be obtained. Therefore, the first subproblem have to be solved as many times as the number of the fleet types to obtained all the feasible flight route sets for each fleet type. Using the distributed computation that is introduced in Section III, the first subproblem for each fleet type can be solved simultaneously. So all the feasible flight route sets for each fleet type can be obtained only once. With the feasible flight route sets, the second subproblem can be modeled as a resource assignment problem which is similar to the model in papers [34], [35]. However, these papers reveal that the resource assignment problem is extremely difficult to solve since it is impossible to enumerate all the feasible flight routes. In this paper, the feasible flight routes are generated by Dang and Ye's method, and they are ordered in the lexicographical order with respect to the original flight routes. Thus, by providing some partial feasible flight routes of the problem (1), the aircrafts reassignment problem can be solved easily using CPLEX Optimizers's Concert Technology. There is no need to compute all the feasible flight routes. Details of the formulation for the aircrafts reassignment problem are given below.

The following notations are used in the formulation.

Indices

- i flight leg index
- r feasible flight route index
- e aircraft fleet type index
- o station index

Sets

- F set of the flight legs
- P^e set of the feasible flight routes for the fleet type e
- Q set of the aircraft fleet types
- S set of the stations

Coefficients

- b_{ir} equal to 1 if the flight leg i is included in the feasible flight route r; 0 otherwise
- h_{or} equal to 1 if the feasible flight route r departures from the station o; 0 otherwise
- g_{or} equal to 1 if the feasible flight route r terminates at the station o; 0 otherwise
- d_r^e cost of reassigning the feasible flight route r to the fleet type e
- q_i cost of canceling the flight leg i
- H_o number of the aircrafts required to departure from the source stations o
- G_o number of the aircrafts required to terminate at the sink stations o
- A^e number of the available aircrafts in the fleet type e

Variables

- y_r^e equal to 1 if the feasible flight route r is reassigned to the fleet type e; 0 otherwise
- z_i equal to 1 if the flight i is canceled; 0 otherwise

Aircrafts Reassignment Mathematical Formulation

$$\text{minimize} \quad \sum_{e\in Q}\sum_{r\in P^e} d_r^e y_r^e + \sum_{i\in F} q_i z_i \tag{3a}$$

subject to (flight cover)

$$\sum_{e\in Q}\sum_{r\in P^e} b_{ir} y_r^e + z_i = 1, \quad \forall i \in F, \tag{3b}$$

(aircrafts balance of source stations)

$$\sum_{e\in Q}\sum_{r\in P^e} h_{or} y_r^e = H_o, \quad \forall o \in S, \tag{3c}$$

(aircrafts balance of sink stations)

$$\sum_{e\in Q}\sum_{r\in P^e} g_{or} y_r^e = G_o, \quad \forall o \in S, \tag{3d}$$

(resource utilization)

$$\sum_{r\in P^e} y_r^e = A^e, \quad \forall e \in Q, \tag{3e}$$

(binary aircraft assignment)

$$y_r^e \in \{0, 1\}, \quad \forall r \in P^e, e \in Q, \tag{3f}$$

(binary cancelation assignment)

$$z_i \in \{0, 1\}, \quad \forall i \in F. \tag{3g}$$

Coefficients b_{ir}, h_{or} and g_{or} can be determined from the feasible solutions of Subsection II-A. The remaining coefficients are inputs derived from the flight schedule.

The objective of (3a) is tominimizes the total cost of each chosen flight route and total cancelation cost of each canceled flight leg. The flight cover constraint (3b) ensures that all the flight legs must be either in a flight route or canceled. The aircrafts balance of source stations constraint (3c) and the aircrafts balance of sink stations constraint (3d) implement the requirement for the number of the aircrafts that are positioned at the beginning and the end of the disruption period for each station. The constraint (3e) ensures that the number of the chosen feasible flight routes for each fleet type is equal to the number of the available aircrafts in the fleet type. The constraints (3f) and (3g) preclude fractional solutions.

III. METHODOLOGY

The problem (1) in Subsection II-A can also be solved by CPLEX as the problem (3) in Subsection II-B, but for a large airline disruption problem it is unlikely to obtain all the feasible flight routes of the problem (1) by CPLEX CP Optimizer in a reasonable amount of time as mentioned by the papers [34], [35]. The experiments from our former works [30], [31], [36] also show that it is impossible to enumerate all the solutions of the problem (1) in a reasonable amount of time.

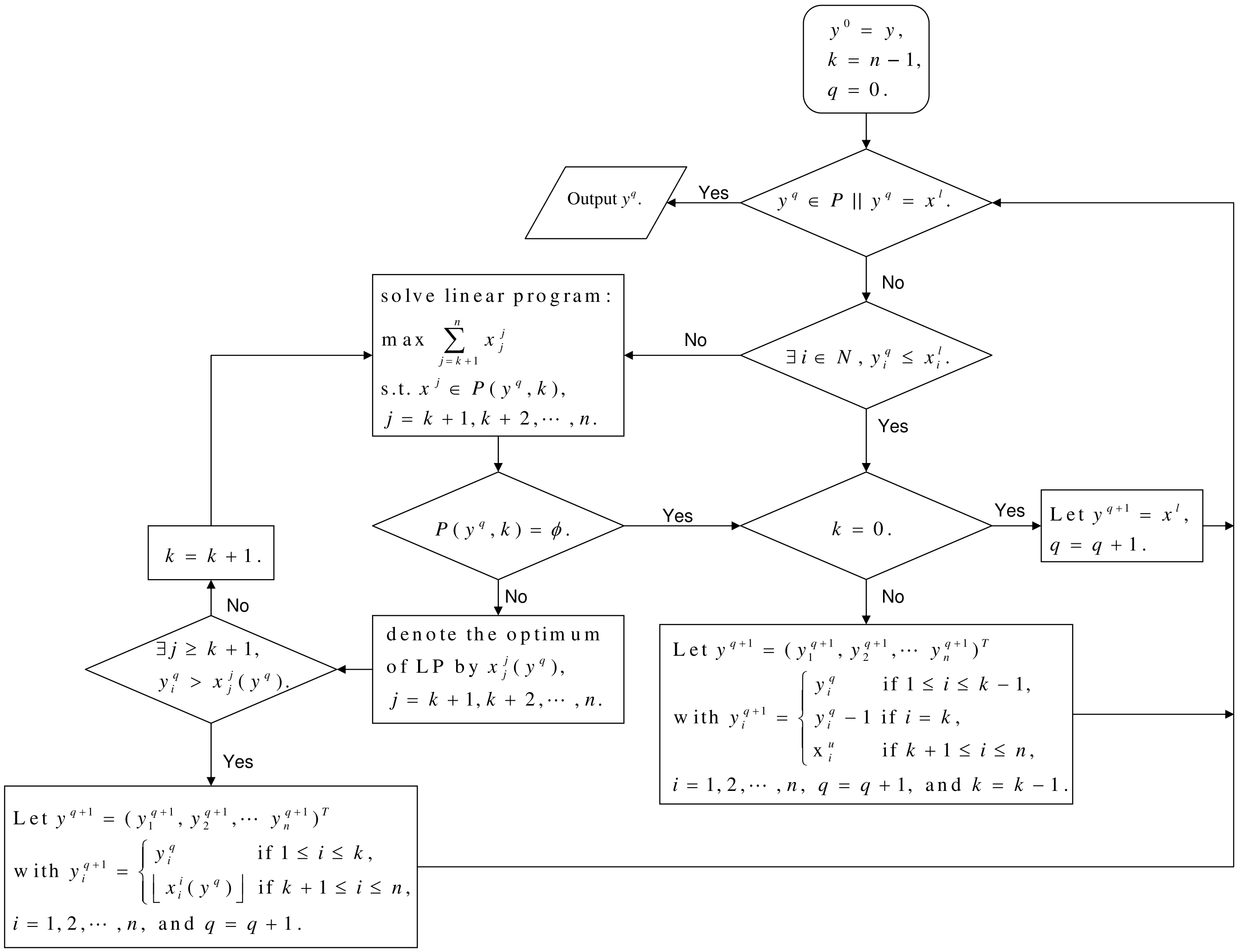


FIGURE 3. Flow diagram of the iterative method.

To tackle this difficulty, the initial seeds cluster division method is introduced to divide the solution space of the problem (1) into several segments, and each divided segment has a unique initial seed. Then Dang and Ye's method is applied to enumerate the feasible solutions of the problem (1) in each divided segment. The obtained feasible solutions of the problem (1) in a segment are only related to the initial seed of the segment, so the computation process of Dang and Ye's method in each divided segment is independent to each other. A distributed computation is proposed so that the computation in each divided segment can be conducted simultaneously. Details of these methods are given below.

A. [28]'s ITERATIVE METHOD

Let $P = \{x \in R^n | Ax + Gw \le b$ for some $w \in R^p\}$, where $A \in R^{m\times n}$ is an $m \times n$ integer matrix with $n \ge 2$, $G \in R^{m\times p}$ an $m \times p$ matrix, and b a vector of R^m.

Let $x^{max} = (x_1^{max}, x_2^{max}, \cdots, x_n^{max})^T$ with $x_j^{max} = max_{x\in P} x_j, j = 1, 2, \cdots, n$, and $x^{min} = (x_1^{min}, x_2^{min}, \cdots, x_n^{min})^T$ with $x_j^{min} = min_{x\in P} x_j, j = 1, 2, \cdots, n$.

Let $D(P) = \{x \in Z^n | x^l \le x \le x^u\}$, where $x^u = \lfloor x^{max} \rfloor$ and $x^l = \lfloor x^{min} \rfloor$.

For $z \in R^n$ and $k \in N_0$, let $P(z, k) = \{x \in P | x_i = z_i, 1 \le i \le k,$ and $x_i \le z_i, k+1 \le i \le n\}$.

Given an integer point $y \in D(P)$ with $y_1 > x_i^l$, [28]'s iterative method is presented in Figure 3. It determines whether there is an integer point $x^* \in P$ with $x^* \le_l y$.

An example given below is used to illustrate the method. Consider a polytope $P = \{x \in \mathbb{R}^3 | Ax \le b\}$ with

$$A = \begin{pmatrix} -1 & 0 & 2 \\ 0 & -2 & 1 \\ -1 & 0 & -2 \\ 1 & 1 & 0 \end{pmatrix} \text{ and } b = \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \end{pmatrix}.$$

It is easy to obtain $x^u = (1, 0, 0)^T$ and $x^l = (-1, -2, -2)^T$. Let $y = x^u$, $y^0 = y$, and $k = 3 - 1 = 2$. $y^1 = (1, -1, 0)$ can be obtained in the first iteration, and $y^2 = (1, -1, -1)$ which is an integer point in P is obtained in the second iteration. An illustration of y^0, y^1, and y^2 can be found in Figure 4.

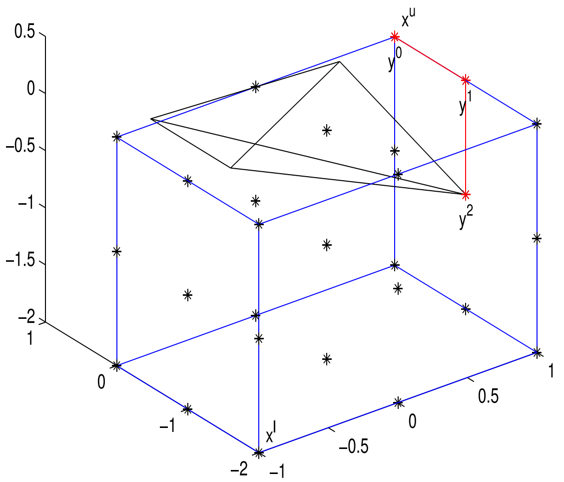


FIGURE 4. An illustration of the iterative method.

Dang and Ye's method is the improvement of the Dang's method [37]. The results obtained by both of the two methods are the same, but the computation processes are different. The idea of Dang and Ye's method to solve the integer programming is to define an increasing-mapping from the lattice into itself. The integer points outside the P are mapped into the first point in P that is smaller than them in the lexicographical order or x^l. All the integer points inside the polytope are fixed points under this increasing mapping. Given an initial integer point, the method either yields an integer point in the polytope or proves no such point exists within a finite number of iterations. With a simple modification all the integer points in P can be obtained sequentially in the lexicographical order. That is to say, for two consecutively obtained points, the one obtained earlier is always larger than the one obtained latter in the lexicographical order, and there is no other feasible point exists between two consecutively obtained points. For more details and proofs about this iterative method, one can refer to the research [28], [37].

B. INITIAL SEEDS CLUSTER DIVISION

Given another polytope P in Figure 5, and a lattice $D(P)$ can be easily constructed by the point x^u and x^l as illustrated in Figure 5. Takeing x^u as y^0 in Figure 3, all the integer points in the polytope P can be enumerated in the lexicographical order using Dang and Ye's method. If providing several initial seed points as the big dots in Figure 5 and taking each of them as y^0 in Figure 3 respectively, the feasible integer points which are smaller than the seed points in the lexicographical order can be obtained using Dang and Ye's method. With a simple modification of Dang and Ye's method, the feasible integer points which are larger than the seed point in the lexicographical order can also be obtained using Dang and Ye's method. So staring from the seed point, the enumeration for the feasible points can be conducted in two directions as in Figure 5. The obtained feasible points form a point cluster around a seed point.

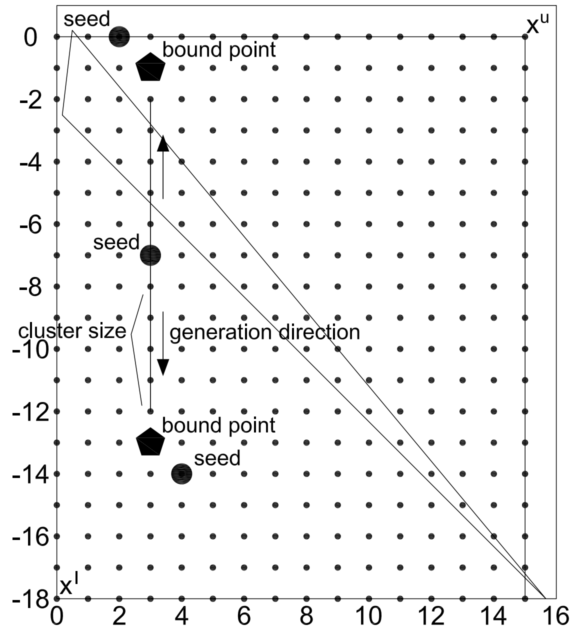


FIGURE 5. Initial seeds cluster division.

Let $x^{b_i} = (x_1^{b_i}, x_2^{b_i}, \cdots, x_n^{b_i})$ for $i = 0, 1, \cdots, m$ denotes a bound point as the pentagon in Figure 5, and $S(P)_i = \{x \in Z^n | x^{b_{i-1}} <_l x \leq_l x^{b_i}\}$ for $i = 1, 2, \cdots, m$ denotes a segment that is defined by two consecutive bound points. Each segment $S(P)_i$ must satisfy the following two conditions:

$$D(P) = \cup_{i=1}^{m} S(P)_i \cup x^l, \tag{4a}$$

$$\emptyset = S(P)_i \cap S(P)_j, \quad \text{for } i, j = 1, 2, \cdots, m, \text{ and } i \neq j. \tag{4b}$$

If the bound points x^{b_i} are calculated, all the segments $S(P)_i$ are obtained and they can be used to enumerate the feasible points in a distributed computation. Let $x^{b_0} = x^l$, $x^{b_m} = x^u$. The calculation process of the remaining bound points is given below.

Let x^{I_i} and $x^{I_{i+1}}$ with $x^l <_l x^{I_i} <_l x^{I_{i+1}} <_l x^u$ be two consecutive initial seeds. The bound point x^{b_i} is in the middle of the x^{I_i} and $x^{I_{i+1}}$ in the lexicographical order. Thus, $x^l <_l x^{I_i} <_l x^{b_i} <_l x^{I_{i+1}} <_l x^u$ holds. Let $B(P)_i = \{x \in Z^n | x^{b_i} \leq_l x <_l x^u\}$, $I(P)_i = \{x \in Z^n | x^{I_i} \leq_l x <_l x^u\}$ and $s^{b_i} = |B(P)_i| = \left\lfloor \frac{|I(P)_i| + |I(P)_{i+1}|}{2} \right\rfloor$. The bound points x^{b_i} for $i = 1, 2, \cdots, m-1$ is given by the Lemma 1.

Lemma 1: Let $x^u = (x_1^u, x_2^u, \cdots, x_j^u, \cdots, x_n^u)$, $x^l = (x_1^l, x_2^l, \cdots, x_j^l, \cdots, x_n^l)$, $p_j = x_j^u - x_j^l + 1$ for $j = 1, 2, \cdots, n$ and $s^{b_i} = |B(P)_i|$ for $i = 1, 2, \cdots, m-1$. Assume that $x^i \leq x^u$. Then, $x^{b_i} = (x_1^{b_i}, x_2^{b_i}, \cdots, x_j^{b_i}, \cdots, x_n^{b_i})$ for $i = 1, 2, \cdots, m-1$ in $D(P)$ is given by the following formula:

$$x_j^{b_i} = \begin{cases} x_j^u - \left\lfloor \dfrac{s^{b_i} \bmod \prod_{k=j}^{n} p_k}{\prod_{k=j+1}^{n} p_k} \right\rfloor, & \text{for } j = 1, 2, \ldots, n-1, \\ x_n^u - s^{b_i} \bmod p_n, & \text{for } j = n. \end{cases} \tag{5}$$

The proof of the Lemma 1 is in our previous work [30], and it is omitted here. Once the bound points are calculated, all the segments are decided. The generation of the feasible points starts from the seed point in each segment using Dang and Ye's method, and the generation goes in both the direction that is lexicographically larger than the seed and the direction that is lexicographically smaller than the seed as illustrated in Figure 5. A parameter cluster size is used to control the number of the feasible points to be obtained in one direction in each segment, so the generation in one of the two directions stops when the number of the obtained feasible points equals the cluster size or a bound point is encountered. The set of the obtained feasible points in each segment form a cluster around the seed point in the segment, and the cluster is a subset of the segment.

This division method is especially devised to solve the feasibility problem (1) in Subsection II-A. When a disruption happens, some of the original flight routes may become irregular and result in losses of revenue. Alternative flight routes of the irregular original flight routes can minimize the losses. So the feasibility problem (1) is formulated to find the alternative flight routes for one fleet type. Take the original flight routes as the initial seed points in Figure 5, and the polytope P and the lattice $D(P)$ can be constructed from the constraints of the problem (1), the computation of the problem (1) for one fleet type can be transformed into the generation of the feasible points in the polytope P in Figure 5 using [28]'s method. Normally there are more than one aircraft in a fleet type, so $D(P)$ can be divided into several segments and the generation of the feasible flight routes from the original flight route in each segment can be conducted simultaneously using [28]'s method in a distributed computation network that is introduced in Subsection III-C.

Dang and Ye's method generates new feasible points that are always lexicographically smaller than the one generated earlier, and it can also generate new feasible points that are always lexicographically larger than the one generated earlier if the generation direction is reversed. So both the feasible flight routes that are lexicographically larger than the original flight routes and the feasible flight routes that are lexicographically smaller than the original flight routes can be found. Each segment $S(P)_i$ can be divided again by another division method proposed in the paper [30] if more computation processors are available.

C. DISTRIBUTED COMPUTATION IMPLEMENTATION

A distributed computation network is constructed by Message Passing Interface (MPI), and it consists of plenty of computers. MPI is used to construct the communication among the processors of each computer. Due to enough computation processors, the distributed computation can increase the computation efficiency and it can be conducted in three different ways as followed:

- The feasibility problem (1) for each fleet type,
- The generation of the feasible flight routes in each divided segment for one feasibility problem (1),
- The linear programming in Dang and Ye's method as illustrated in Figure 3

The feasible flight routes are generated only for one fleet type when the feasibility problem (1) is solved once. So the problem (1) have to be solved as many times as the number of the fleet types to obtained all the feasible flight route sets for all the fleet types. The problem (1) for each fleet type differs in the matrices of polytope Ps that are constructed from the constraints (1b) to (1m) and the original flight routes. After the matrices and the original flight routes of each fleet type are sent to each participated computers in the computation network, the distributed computation can be started. The communication of data can be easily done by the send operation MPI_SEND and the receive operation MPI_RECV from MPI, and the control of all the participated computers can also be easily done by MPI. Details of the usage of MPI can be found in "A Message-Passing Interface Standard" [38].

The $D(P)$ of each feasibility problem (1) can be divided into several segments, and these segments satisfy the condition (4). Besides, the obtained feasible flight routes of the problem (1) in a segment is only related to the original flight route of the segment. So the computation process of Dang and Ye's method in each divided segment is independent to each other and it can be conducted simultaneously in each computation processor. Details of the distributed computation among segments of one feasibility problem (1) can be found in the paper [30].

The linear programming in Dang and Ye's method aims at maximizing the sum of a series of variables. Since these variables are independent to each other, they can be maximized individually and the results are the same. Thus, the linear programming in Figure 3 can be reformulated as followed:

$$\text{maximize } x_j^j, \quad \text{for } j = k+1, k+2, \cdots, n, \tag{6a}$$

$$\text{subject to } x^j \in P(y^q, k). \tag{6b}$$

The number of the linear programming increases from 1 to $n-k$, and these $n-k$ linear programming can be solved simultaneously in the distributed computation network if there are enough computation processors. It can also be done by MPI easily, and the implementations are conducted on a simple problem that has few variables. No implementation on the problems in Section IV is conducted due to the limited computation processors. The distributed computation of the linear programming can indeed save a lot of computation time for large-degree problems with enough computation processors provided.

In an implementation, the previous two ways of the distributed computations can be treated together as illustrated in Figure 6. Since each segment is divided based on an original flight route of an aircraft. The total segments for all the aircrafts in all the fleet types can be obtained first. The generation of the feasible flight routes starts from the original flight routes within the bound points in each segment. So for

TABLE 4. Comparison with [39].

Method	Cluster size	Delayed flights	Total delay minutes	Flight swaps	Unaltered flight routes
Wu	2	0	0	9	24
	5	0	0	8	24
	10	0	0	5	26
Babic	\	2	255	3	26
		3	361	5	25
		2	483	5	25
Dispatchers' solution	\	0	0	5	26

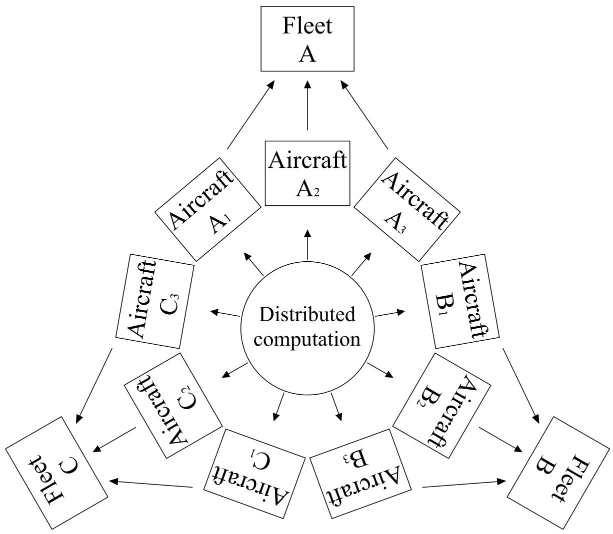


FIGURE 6. Implementation of distributed computation.

the generation of the feasible flight routes for the aircrafts in one fleet type, the differences are the bound points that define the segments and the original flight routes of the aircrafts. For the generation of the feasible flight routes for all the aircrafts in all the fleet types, the differences are the bound points that define the segments, the original flight routes of the aircrafts and the matrices of polytope Ps. The distributed computation is implemented by sending the bound points, the original flight routes and the matrices of polytope Ps to each computation processors in the distributed computation network.

IV. COMPUTATIONAL EXPERIENCE

Airline flight schedules that appeare in the paper [16], [39] are used to evaluate our approach to the airline disruption problems, and details of the schedules can be found in the research [16], [39]. The distributed computation network in which is constructed by MPI consisting of 3 different computers, and one computer can establish 16 threads simultaneously to compute and the other two can establish 2 threads. So 20 segments can be processed simultaneously at most. All the programs are coded in C++, and CPLEX Concert Technology is used to solve the linear programming in Dang and Ye's method. The version of the Cplex used is 12.6.1. First the problem (1) in Subsection II-A is solved using [28]'s method. Then all these feasible flight routes are used to generate a solution for the problem (3) in Subsection II-B. CPLEX Optimizers's Concert Technology is used to solve the problem (3).

In our previous work [30], [31], numerical results have showed that Dang's method and Dang and Ye's method outperform CPLEX CP Optimizer when they are used to solve the problem(1) respectively. So no comparison is made between Dang and Ye's method and CPLEX CP Optimizer in this paper. We mainly focus on the comparisons between present results and the results in the paper [16], [39].

A. EUROPEAN MID-SIZE AIRLINE RESULTS

Paper [39] employs a European mid-size airline schedule to demonstrate their approach. The schedule consists of 29 aircrafts from 10 different fleet types that are Airbus 319, 320, 321, 322 and 32C, Boeing 736, 738, 739 and 73G and Fokker F70. Each of the fleet has a different passenger capacity. 126 flight legs are served by all the aircrafts between 43 stations. The lattice $D(P)$ of the problem (1) is divided into 29 segments and each of the segment has a unique seed point which is the original flight route and a unique pair of bound points. For the segments whose corresponding aircrafts are in the same fleet type, they share the same matrix of polytope P. Since only 20 segments can be processed simultaneously, the last 9 segments can be started to compute after any one of the first 20 segments is finished.

A disruption given in the research [39] happens when the aircraft 321-YYLBC is breakdown and under repair from 8:00 AM until 11:00 PM. So the disruption lasts for 15 hours. The flight legs XX863 VIE-CAI, XX864 CAI-VIE and XX127 VIE-FRA which are originally assigned to the aircraft 321-YYLBC are affected by the disruption. These flight legs may be canceled if no recovery plan is made. The cancelation of the flight leg XX127 VIE-FRA causes aircrafts unbalance at the airport FRA at the end of the schedule. So a recovery plan is needed to make sure that there is an aircraft substituting the aircraft 321-YYLBC and terminating at the airport FRA at the end of the schedule. Besides, the cancelations of the flight legs result in huge losses of revenue, so the recovery plan also aims at minimizing the losses.

Table 4 and Table 5 are the solutions obtained by our approach and the comparison with the solutions that are given in the paper [39]. Row two to row four of Table 4 are the solutions of the problem (3) obtained by our approach, and each solution is a recovery plan of the airline disruption problem. The solutions are computed by the feasible flight

TABLE 5. Details of solutions that are obtained by our approach.

<table>
<tr><th>Method</th><th>Cluster size</th><th>Details of solutions</th></tr>
<tr><td rowspan="3">Wu</td><td>2</td><td>321-YYLBA: XX863, XX864, XX127
738-YYLNR: XX1425, XX9741, XX9742, XX1426, XX9767, XX9768
738-YYLNS: XX9863, XX9864, XX125, XX126
320-YYLBT: XX458, XX801, XX802, XX417, XX418</td></tr>
<tr><td>5</td><td>321-YYLBA: XX863, XX864, XX127
738-YYLNR: XX1425, XX9741, XX9742, XX1426, XX307
320-YYLBO: XX308, XX125, XX126
320-YYLBT: XX458, XX801, XX802, XX417, XX418</td></tr>
<tr><td>10</td><td>321-YYLBA: XX863, XX864, XX127
738-YYLNR: XX1425, XX9741, XX9742, XX1426, XX417, XX418</td></tr>
</table>

routes that are generated in the problem (II-A). The parameter cluster size in the second column is used to control the number of the feasible flight routes that are generated in both of the two directions in each segment. So the total number of the feasible flight routes that are obtained for each available aircraft equals to the double of the cluster size. Column three and column four are number of the delayed flight legs and the total delay time in minute respectively. Column five are the number of the flight legs that are swapped with other flight routes' flight legs. Column six are the number of the flight routes that are not affected by the disruption, and they are flew as the plans in the original schedule.

The cluster size is enlarged to test the impact of the number of the feasible flight routes on the solution of the problem (3). Larger cluster size can find a better solution of the disruption problem, because more feasible flight routes that are involved in the constitution of the recovery plan means more possibilities that a better recovery plan with less flight swaps and more unaltered flight routes can be produced. When the cluster size increases to 10, the solution which is the same as the dispatchers' solution in row eight is found. The recovery plan that is obtained under the cluster size 10 is what the airline dispatchers would implement.

Row five to row seven of Table 4 are solutions that are computed by the paper [39]. There are delayed flight legs in each of the recovery plan they produced. Although there are less flight swaps in their solutions, the amount of the total delayed flight time brings a huge losses of revenue to airline. The airline dispatchers would not implement any of these solutions that are computed by the paper [39]. The dispatchers' solution that would be implemented by the dispatchers in row eight is given not through computation. It is obtained by modifying the solution in the row seven of Table 4.

Details of solutions that are obtained by our approach can be found in Table 5. The aircrafts appeared in Table 5 fly the flight legs that are reassigned to them, and the remaining aircrafts that are not appeared fly their original flight route in the recovery plan. The solutions without delayed flight legs can be found only under the cluster size 2, that means only 4 feasible flight routes are needed for each available aircraft to produce the recovery plan. 20 feasible flight routes are needed for each available aircraft to produce the recovery plan that the airline dispatchers would implement. Also all the solutions that are computed by the paper [39] in row five to row seven of Table 4 can be constituted using the feasible flight routes that are obtained under the cluster size 10. This is done by setting the parameter "SolnPoolAGap" of CPLEX Optimizerss Concert Technology to a proper value. A larger value of the parameter "SolnPoolAGap" means more non-optimal solutions can be obtained. Also non-optimal solutions which deviate too much from the optimum can be found by setting a larger "SolnPoolAGap".

Our approach is more effective than the method [39] when solving this European mid-size airline schedule disruption problem considering the dispatchers' solution can be computed by our approach but it can not be computed by the approach proposed by the paper [39].

B. SWEDISH DOMESTIC AIRLINE RESULTS

The research [16] use two flight schedules, $s1$ and $s2$, from Swedish domestic airline to illustrate their approach. The characteristics of the two schedules are given in Table 6. The two schedules are disrupted in two different ways, a and b, that are given in [16]. a is disrupted by an unavailability of an aircraft and b is disrupted by an imposed delay on several flight legs. So there are four disruption problems $s1a$, $s1b$, $s2a$ and $s2b$. Causes of the disruptions are given in Table 7.

TABLE 6. Swedish domestic airline schedule characteristics.

	$s1$	$s2$
Number of aircraft	13	30
Number of aircraft types	2	5
Number of flight legs	98	215
Number of airports	19	32

The paper [16] aims at maximizing the revenue of airline when solving the disruption problems, and they use weights to calculate the costs of the disruptions. So the objective function (3a) of the problem (3) is modified to maximize the revenue of the airline.

The following notations are used in the formulation of the modified objective function.

Weights

- can cancelation weight
- sw_t weight for swaps between the flight legs in different fleet types
- sw weight for swaps between the flight legs in the same fleet types
- del delay weight

TABLE 7. Causes of the disruptions.

Disruption problems	Causes of the disruptions
$s1a$	An unavailability of aircraft for 5 hours
$s1b$	Imposed delays on two flight legs for 25 and 30 minutes respectively
$s2a$	An unavailability of aircraft for 6 hours
$s2b$	Imposed delays on four flight legs for 15, 20, 30 and 40 minutes respectively

Coefficients

- $\boldsymbol{st_r}$ number of swaps between the flight legs in different fleet types in the feasible flight route r
- $\boldsymbol{s_r}$ number of swaps between the flight legs in the same fleet types in the feasible flight route r
- $\boldsymbol{d_r}$ delay in passenger minutes in the feasible flight route r

With the weights and new coefficients, the modified objective function is formulated below:

$$\text{maximize} \sum_{e \in Q} \sum_{r \in P^e} (can \times \sum_{i \in F} b_{ir} c_i - sw\_t \times \boldsymbol{st_r} - sw \times \boldsymbol{s_r} - del \times \boldsymbol{d_r}) y_r^e \quad (7)$$

So the aircrafts reassignment problem in Subsection II-B is to maximize the objective (7) under the constraints (3b) to (3g) for these disruptions problems from the Swedish domestic airline. To be convenient for the comparison, the notations of the weights and the new coefficients are the same as those in the paper [16]. But the notations of the new coefficients are in bold in order to distinguish them from the notations that have been used before. For example, $\boldsymbol{d_r}$ is different from d_r^e that appears in the problem (3) in Subsection II-B.

The objective (7) is the sum of the revenues of the feasible flight routes. The revenue of each feasible flight route is calculated by subtracting the swap cost $sw\_t \times \boldsymbol{st_r}$, $sw \times \boldsymbol{s_r}$ and delay cost $del \times \boldsymbol{d_r}$ from the cancelation weight can multiplied by the actual passenger load $\sum_{i \in F} b_{ir} c_i$ on the feasible flight route. The coefficients $\boldsymbol{st_r}$ and $\boldsymbol{s_r}$ can be inspected from each feasible flight route. For a delayed flight leg, the delay in passenger minutes equals the actual passenger load on the delayed flight leg multiplied by the delay time of the delayed flight leg. So the coefficient $\boldsymbol{d_r}$ is calculated by summing the delays in passenger minutes of each delayed flight leg in the feasible flight route r.

The paper [16] provides several different weight settings for each of the four disruption problems $s1a$, $s1b$, $s2a$ and $s2b$ as in Table 8. Thus more problem instances can be generated from the original four disruption problems. $s1a1$ uses flight schedule $s1$ with the disruption a and weight setting 1 which sets can to 20, sw_t to 100, sw to 20 and del to 1 to generate a disruption problem. A unique solution with different characteristics can be obtained with each weight setting, and

TABLE 8. Sets of weights.

Problem	Weights			
	can	sw_t	sw	del
$s1a1$	20	100	10	1
$s1a2$	20	1000	10	1
$s1a3$	100	1000	10	1
$s1a4$	100	100	10	1
$s1b1$	20	100	10	1
$s1b2$	20	100	100	1
$s1b3$	20	100	400	1
$s2a1$	20	100	10	1
$s2a2$	20	1000	10	1
$s2a3$	100	1000	10	1
$s2b1$	20	100	10	1
$s2b2$	20	100	10	10
$s2b3$	20	100	50	1

TABLE 9. Results of $s1$.

Problem	Cluster size	Obj	Characteristics			
			$\boldsymbol{c}$	$\boldsymbol{st}$	$\boldsymbol{s}$	$\boldsymbol{d}$
$s1a1$	3	42660	32	0	0	1420
	10	43760	46	0	4	0
	50	43800	25	4	2	0
	2200	**43840**	**30**	**2**	**8**	**0**
$s1a2$	3	42660	32	0	0	1420
	10	**43760**	**46**	**0**	**4**	**0**
$s1a3$	**3**	**218980**	**32**	**0**	**0**	**1420**
$s1a4$	3	218980	32	0	0	1420
	50	**220680**	**25**	**4**	**2**	**0**
$s1b1$	5	43140	0	0	0	1580
	25	43340	0	0	5	1330
	75	43620	0	0	15	950
	100	44445	0	0	10	175
	125	***44650***	***0***	***0***	***7***	***0***
$s1b2$	5	43140	0	0	0	1580
	100	43870	0	0	6	250
	125	**44070**	**0**	**0**	**4**	**250**
$s1b3$	**5**	**43140**	**0**	**0**	**0**	**1580**

all of the 13 generated problems are used to evaluate our approach.

Table 9 and Table 10 are the results of the problems generated from the schedule $s1$ and $s2$ respectively. Obj is the solution of the objective function (7) under the constraints (3b) to (3g). $\boldsymbol{c}$ is the number of the canceled passengers in the solution. $\boldsymbol{st}$ is the number of swaps between the flight legs in different fleet types in the solution. $\boldsymbol{s}$ is the number of swaps between the flight legs in the same fleet types in the solution. $\boldsymbol{d}$ is the delay in passenger minutes in the solution.

In Table 9, the solution of each problem generated from schedule $s1$ under different cluster size is given. The cluster size is increased to find to a better solution of each problem since more feasible flight routes can constitute a better solution. Once a solution which is equal to or better than

TABLE 10. Results of $s2$.

Problem	Cluster size	Obj	Characteristics			
			c	st	s	d
$s2a1$	260	68480	29	2	4	0
$s2a2$	5	68000	65	0	0	0
$s2a3$	1500	341865	29	0	12	1615
$s2b1$	1500	69080	0	0	12	100
$s2b2$	1500	68720	23	0	12	0
$s2b3$	***10***	***68950***	***0***	***2***	***1***	***100***

that computed by the research [16], the cluster size stoppes increasing. The row in bold is the best solution of each problem, and a solution which is better than that in the research [16] is found for the problem $s1b1$ in the row marked in bold and italic. It is obtained under the cluster size 125, and the objective value is larger than that in the paper [16] by 50. The details of the solution are offered bellow. The 5 aircrafts fly the flight legs that are reassigned to them, and the remaining aircrafts that are not appeared below fly their original flight routes in the recovery plan.

- SELEA: 22, 23, 77, 78, 24, 25, 26, 27, 28, 29, 30
- SELEC: 42, 43, 44, 52, 53
- SELED: 46, 47, 48, 49, 50, 51, 45
- SELEZ: 76, 89, 90, 79, 80, 81, 82
- SELIP: 88, 91, 92, 93, 94, 95, 96, 97

In Table 10, the final solution of each problem generated from schedule $s2$ is given. The first 5 solutions equal those computed by the research [16]. The solution of the last problem $s2b3$ is better than that of paper [16] by 50, and it is computed only under the cluster size 10. Details of the solution of the problem $s2b3$ are given below.

- SEISY: 37, 40, 41, 42, 43, 44 ,45, 46
- SELED: 120, 121, 122, 123, 124, 125, 126, 127
- SELIN: 200, 201, 202, 203, 204, 205, 206, 128
- SELIP: 207, 38, 39, 208, 209, 210, 211, 212, 213, 214

It can be seen that present approach is effective to solve the airline disruption problems. Not only the solutions that are equal to the solutions in literature can be computed, but also the solutions that are better than the solutions in literature can be found for some problems. It is because each feasible flight route is generated from an original flight route, and the feasible flight routes can be obtained sequentially in the lexicographical order using Dang and Ye's method. The feasible flight routes have relations with the original flight routes. Some of the flight legs in the feasible flight routes are different from those in the original flight routes, while others are the same. The number of the different flight legs in the feasible flight routes increase as the generation for the feasible flight routes go on. Sometimes the feasible flight routes with more flight legs that are different from those in the original flight routes can constitute a better solution of the disruption problem, and more feasible flight routes have to be generated before the feasible flight routes with more flight legs that are different from those in the original flight routes can be obtained. So the quality of the solutions are related to the number of the feasible flight routes that are controlled by the cluster size. Some problems only need a few feasible flight routes to obtain a better solution of the disruption problem, but some other problems may need much more. Basically speaking, a much better solution of the disruption problem can be obtained by providing more feasible flight routes. In an actual implementation, the cluster size can be set to a relatively larger value. And the computation of the disruption problem can be conducted periodically. For example, the aircrafts reassignment problem (3) in Subsection II-B can be solved once 100 new feasible flight routes are obtained. The computation stops when the solution satisfies the airline dispatchers.

V. CONCLUSION

A novel modified TSP model is formulated to generate feasible flight routes for producing a recovery plan for the airline disruption problems. A distributed computation network is constructed to solve the modified TSP model using Dang and Ye's method, and the feasible flight routes for all the aircrafts in all the fleet types can be generated simultaneously in the distributed computation. Dang and Ye's method ensures that the obtained feasible flight routes are in the lexicographical order automatically, and there does not exist other feasible flight route between two consecutively obtained feasible flight routes. This characteristic enables the generated feasible flight routes to constitute a recover plan easily, and only partial feasible flight routes are needed. If more feasible flight routes are provided, a much better recover plan can be constituted. In the evaluation of present approach, not only the same solutions as those in literature can be computed, but also solutions that are better than those in literature can be found.

In the future work, the distributed computation network will be expanded by incorporating with the third way of the distributed computation that is mentioned in Subsection III-C. For large problems, the (6) which is the most computationally expensive part can be computed in less time. Thus, the solutions of the disruption problems can be obtained more efficiently.

COMPETING INTERESTS

The authors declare that they have no competing interests.

AUTHOR's CONTRIBUTIONS

All authors read and approved the final manuscript.

ACKNOWLEDGMENTS

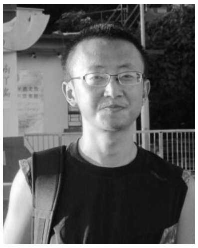
The authors are very grateful to the editor and reviewers for their valuable comments and suggestions. *(Zhengtian Wu, Benchi Li, and Chuangyin Dang contributed equally to this work.)*

REFERENCES

[1] J. Clausen, A. Larsen, J. Larsen, and N. J. Rezanova, "Disruption management in the airline industry—Concepts, models and methods," *Comput. Oper. Res.*, vol. 37, pp. 809–821, May 2010.

[2] A. Abdelghany and K. Abdelghany, *Modeling Applications in the Airline Industry*. Farnham, U.K.: Ashgate, 2010.
[3] S.-C. Huang, "Airline schedule recovery following disturbances: An organizationally-oriented decision-making approach," Ph.D. dissertation, Dept. Civil Environ. Eng., Univ. California, Berkeley, Berkeley, CA, USA, 2005.
[4] M. D. Clarke and B. C. Smith, "Impact of operations research on the evolution of the airline industry," *J. Aircraft*, vol. 41, no. 1, pp. 62–72, 2004.
[5] *Statistical Communiqué on Civil Aviation Industry Development in 2010, Air Transport & Business*, Civil Aviation Administration of China, Beijing, China, 2011, pp. 19–24.
[6] Z. Jinfu, *Air Transportation Operation*. Xi'an, China: Northwestern Polytech. Univ. Press, 2009.
[7] CAAC News. (2012). *Liu Shaoyong: The Airline Operating Cost Increases by CNY1,000 for Each Minute' Delay*. [Online]. Available: http://editor.caacnews.com.cn/mhb/html/2012-03/28/content_93332.htm
[8] EUROCONTROL. (2012). *Performance Review Report: An Assessment of Air Traffic Management in Europe During the Calendar Year 2011*. [Online]. Available: http://www.eurocontrol.int/sites/default/files/content/documents/single-sky/pru/publications/prr/prr-2011-draft.pdf, 2012.
[9] CODA. (2012). *Delays to Air Transport in Europe*. [Online]. Available: http://www.eurocontrol.int/documents/coda-digest-annual-2011
[10] Air Transport Association. (2011). *Annual and Per-Minute Cost of Delays to U.S. Airlines*. [Online]. Available: http://www.airlines.org/Pages/Annual-and-Per-Minute-Cost-of-Delays-to-U.S.-Airlines.aspx
[11] A. Niehues, S. Belin, T. Hansson, R. Hauser, M. Mostajo, and J. Richter, *Punctuality: How Airlines can Improve on-Time Performance*. McLean, VA, USA: Booz Allen & Hamilton Inc., 2001.
[12] D. Teodorović and S. Guberinić, "Optimal dispatching strategy on an airline network after a schedule perturbation," *Eur. J. Oper. Res.*, vol. 15, no. 2, pp. 178–182, 1984.
[13] D. Teodorović and S. Guberinić, "Model for operational daily airline scheduling," *Transp. Planning Technol.*, vol. 14, no. 4, pp. 273–285, 1990.
[14] D. Teodorović and S. Guberinić, "Model to reduce airline schedule disturbances," *J. Transp. Eng.*, vol. 121, no. 4, pp. 324–331, 1995.
[15] J. M. Rosenberger, E. L. Johnson, and G. L. Nemhauser, "Rerouting aircraft for airline recovery," *Transp. Sci.*, vol. 37, no. 4, pp. 408–421, 2003.
[16] T. Andersson and P. Värbrand, "The flight perturbation problem," *Transp. Planning Technol.*, vol. 27, no. 2, pp. 91–117, 2004.
[17] S. Yan, C. Tang, and C. Shieh, "A simulation framework for evaluating airline temporary schedule adjustments following incidents," *Transp. Planning Technol.*, vol. 28, no. 3, pp. 189–211, 2005.
[18] M. Løve, K. R. Sørensen, J. Larsen, and J. Clausen, "Disruption management for an airline—Rescheduling of aircraft," in *Applications of Evolutionary Computing* (Lecture Notes in Computer Science), S. Cagnoni, J. Gottlieb, E. Hart, M. Middendorf, and G. Raidl, Eds. Berlin, Germany: Springer, 2002, pp. 289–300.
[19] M. F. Arguello, J. F. Bard, and G. Yu, "Models and methods for managing airline irregular operations," in *Operations Research in the Airline Industry* (International Series in Operations Research & Management Science), vol. 9, G. Yu, Ed. New York, NY, USA: Springer, 1998, pp. 1–45.
[20] J. Bard, G. Yu, and M. Arguello, "Optimizing aircraft routings in response to groundings and delays," *IIE Trans.*, vol. 33, pp. 931–947, Apr. 2001.
[21] L. Lettovský, "Airline operations recovery: An optimization approach," Ph.D. dissertation, Dept. Ind. Syst. Eng., Georgia Inst. Technol., Atlanta, GA, USA, 1997.
[22] N. Kohl, A. Larsen, J. Larsen, A. Ross, and S. Tiourine, "Airline disruption management—Perspectives, experiences and outlook," *J. Air Transp. Manage.*, vol. 13, no. 3, pp. 149–162, 2007.
[23] T. Liu, C. Jeng, and Y. Chang, "Disruption management of an inequality-based multi-fleet airline schedule by a multi-objective genetic algorithm," *Transp. Planning Technol.*, vol. 31, no. 6, pp. 613–639, 2008.
[24] B. Munyazikwiye, H. R. Karimi, and K. Robbersmyr, "Optimization of vehicle-to-vehicle frontal crash model based on measured data using genetic algorithm," *IEEE Access*, vol. 5, pp. 3131–3138, 2017.
[25] Y. Wei, J. Qiu, and H. R. Karimi, "Reliable output feedback control of discrete-time fuzzy affine systems with actuator faults," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 64, no. 1, pp. 170–181, Jan. 2017.
[26] W. Ji, A. Wang, and J. Qiu, "Decentralized fixed-order piecewise affine dynamic output feedback controller design for discrete-time nonlinear large-scale systems," *IEEE Access*, vol. 5, pp. 1977–1989, 2017.
[27] W. Xing, Y. Zhao, and H. Karimi, "Convergence analysis on multi-AUV systems with leader-follower architecture," *IEEE Access*, vol. 5, pp. 853–868, 2017.
[28] C. Dang and Y. Ye, "A fixed point iterative approach to integer programming and its distributed computation," *Fixed Point Theory Appl.*, vol. 2015, no. 1, p. 182, Oct. 2015.
[29] B. Li, C. Dang, and J. Zheng, "Solving the large airline disruption problems using a distributed computation approach to integer programming," in *Proc. Int. Conf. Inf. Sci. Technol.*, 2013, pp. 444–450.
[30] Z. Wu, B. Li, C. Dang, F. Hu, Q. Zhu, and B. Fu, "Solving long haul airline disruption problem caused by groundings using a distributed fixed-point computational approach to integer programming," *Neurocomputing*, to be published. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0925231217309943?via%3Dihub
[31] Z. Wu, B. Li, and C. Dang, "Solving the large airline disruption problems using a distributed computation approach to integer programming," in *Proc. Workshop Paper*, Mar. 2017, pp. 444–450.
[32] B. G. Thengvall, "Models and solution techniques for the aircraft schedule recovery problem," Ph.D. dissertation, Graduate Program Oper. Res. Ind. Eng., Univ. Texas, Austin, Austin, TX, USA, 1999.
[33] L. A. Wolsey and G. L. Nemhauser, *Integer and Combinatorial Optimization* (Series in Discrete Mathematics and Optimization). Hoboken, NJ, USA: Wiley, 1988.
[34] M. F. Argüello, "Framework for exact solutions and heuristics for approximate solutions to airlines' irregular operations control aircraft routing problem," Ph.D. dissertation, Dept. Mech. Eng., Univ. Texas, Austin, Austin, TX, USA, 1997.
[35] M. F. Argüello, J. F. Bard, and G. Yu, "A grasp for aircraft routing in response to groundings and delays," *J. Combinat. Optim.*, vol. 1, pp. 211–228, Oct. 1997.
[36] Z. Wu, C. Dang, H. R. Karimi, C. Zhu, and Q. Gao, "A mixed 0-1 linear programming approach to the computation of all pure-strategy Nash equilibria of a finite n-person game in normal form," *Math. Problems Eng.*, vol. 2014, Mar. 2014, Art. no. 640960.
[37] C. Dang, "An increasing-mapping approach to integer programming based on lexicographic ordering and linear programming," in *Proc. 9th Int. Symp. Oper. Res. Appl. (ISORA)*, Chengdu-Jiuzhaigou, China, Aug. 2010, pp. 55–60.
[38] Message Passing Interface Forum. (2009). *MPI: A Message-Passing Interface Standard, Version 2.2*. [Online]. Available: http://www.mpi-forum.org/docs/mpi-2.2/mpi22-report.pdf
[39] O. Babic, M. Kalic, D. Babic, and S. Dozic, "The airline schedule optimization model: Validation and sensitivity analysis," *Procedia-Social Behav. Sci.*, vol. 20, pp. 1029–1040, Jan. 2011.

ZHENGTIAN WU received the B.Sc. degree in mechanical manufacturing and automation from the Hefei University of Technology, China, in 2008, and the Ph.D. degrees in operations research from the University of Science and Technology of China and City University of Hong Kong in 2014. He is currently a Lecturer with the Suzhou University of Science and Technology, Suzhou, China. His research interests include Nash equilibrium, mixed integer programming, approximation algorithm, and distributed computation.



BENCHI LI received the B.Sc. degree in mechanical manufacturing and automation from the University of Science and Technology of China, China, in 2007, and the Ph.D. degrees in operations research from the University of Science and Technology of China and City University of Hong Kong in 2013. He is currently a Researcher with China National Petroleum, China. His research interests include approximation algorithm and distributed computation.

CHUANGYIN DANG received the B.Sc. degree in computational mathematics from Shanxi University, China, in 1983, the M.Sc. degree in applied mathematics from Xidian University, China, in 1986, and the Ph.D. *(Cum Laude)* degree in operations research and economics from Tilburg University, The Netherlands, in 1991. He is currently a Professor with the Department of Systems Engineering and Engineering Management, City University of Hong Kong, China. He has published over 100 papers in journals. His research interests include computational optimization, logistics, supply chain management, and modeling in economics.