

Research Article

Improved Ant Colony Optimization for the Operational Aircraft Maintenance Routing Problem with Cruise Speed Control

Qing Zhang,¹ Felix T. S. Chan ,² and Xiaowen Fu ¹

¹Department of Industrial and Systems Engineering, The Hong Kong Polytechnic University, Hung Hum, Hong Kong

²Department of Decision Sciences, Macau University of Science and Technology, Avenida Wai Long, Taipa, Macau

Correspondence should be addressed to Felix T. S. Chan; tschan@must.edu.mo

Received 27 March 2023; Revised 20 April 2023; Accepted 10 May 2023; Published 20 May 2023

Academic Editor: Zhixiang Fang

Copyright © 2023 Qing Zhang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The operational aircraft maintenance routing problem (OAMRP) plays a critical part in producing considerable cost reductions for airlines, since its solution directly influences the number of operating leased aircraft. To reduce the quantity of required aircraft, adopting cruise speed control in OAMRP is a good strategy. In this paper, we investigate the OAMRP with cruise speed control. The objective is to minimize the required quantity of aircraft by finding the optimal aircraft routes through cruise time optimization. The focus is on solving two issues simultaneously: (i) optimization of cruise times and (ii) determination of aircraft routes. Since the combination of two intricate sets of decisions poses significant methodological challenges, the difficulty lies in how to efficiently solve it. Accordingly, the goal of this study is twofold: (i) to design a preprocessing step to reduce the network size and (ii) to develop an improved ant colony optimization (IACO) algorithm with a new state transition mechanism to provide the guidance for cruise times optimization and a new pheromone updating mechanism to enhance the search efficiency and precision. Using data from the Bureau of Transportation Statistics (BTS), we demonstrate the computational efficiency of the preprocessing step and the IACO algorithm.

1. Introduction

1.1. Background and Motivation. The airline scheduling plan plays an indispensable role in both the smooth operation and market competitiveness of airlines [1–3]. Given its large size and high complexity, it has been disaggregated into four sequentially solved stages in practice. They include the flight scheduling problem (FSP), the fleet assignment problem (FAP), the aircraft maintenance routing problem (AMRP), and the crew scheduling problem (CSP) [4–9]. Among them, we are particularly interested in the AMRP, which seeks to identify the assignment of available aircraft to fly full range of flights while simultaneously complying with maintenance requirement. These requirements necessitate that an aircraft receives a maintenance check before exceeding the limitation for accumulated flying hours, takeoffs, and flying days. There are four types of maintenance checks, denoted by the letters A, B, C, and D, respectively, each with a different scope, frequency, and duration. A-checks are taken into

account in the AMRP because they are the most frequently performed [4–9]. Inefficient flight connection or inappropriate maintenance arrangement may cause unnecessary ground time for an aircraft, which can be considered as a loss for an airline. Hence, research and development on the AMRP with the objective of maximizing aircraft utilization have attracted increasing attention.

Notably, the importance of improving aircraft utilization is becoming more prominent in the context of the growing trend of aircraft leasing. It is known that purchasing an aircraft requires huge up-front capital investment. Moreover, airlines must cover the depreciation costs during the useful life of an aircraft. To reduce the financial burden, airlines have shifted to aircraft leasing as their main channel for fleet acquisition [10–12]. In the past decades, the share of the leased aircraft has increased exponentially from 5% in 1970 to 51% in 2021 [13]. Additionally, it is anticipated that the global aircraft leasing market will even reach a value of 29,580,000,000\$ in 2029 [14].

Among all types of aircraft lease, the operating lease is the most favourable type for airlines due to its operational flexibility [11, 15]. It is adopted to acquire additional fleet capacity for a short period of time, thereby enabling the fleet composition and size to be managed closely to the actual operation [10]. In this connection, airlines can make more accurate aircraft leasing decisions by consulting their operational schedules. Based on interviews with employees of low-cost airlines (i.e., China West Air and Spring Airlines) whose fleet is predominantly on operating leases, the number of leased aircraft is determined by the required quantity of aircraft to cover all flights in an airline's network, which is decided in the operational AMRP (OAMRP). It is worth noting that the rental costs of operating lease are higher when compared with other leasing alternatives and increase with the rising demand for operating leases in the market [16], which lead to further shrunk of the already low profit margin of airlines. Improving the usage of aircraft and thereby reducing the quantity of required aircraft can significantly cut airline costs, which motivates the authors to conduct this research.

In this paper, we investigate a new variant of the OAMRP, i.e., the OAMRP incorporating cruise speed control, which was first proposed by Zhang et al. [17] to decrease the required quantity of aircraft. By changing cruise times of flights in the OAMRP, the unconnectable flights can be transformed into connectable ones, resulting in a larger solution space and hence the opportunity for more efficient routing. Nevertheless, the combination of two intricate sets of decisions, namely, cruise time decisions and aircraft route decisions, poses significant methodological challenges.

Since the AMRP is NP-hard, heuristics and meta-heuristics are much more preferred [18–31]. Among these approaches, ACO is particularly appropriate for the OAMRP in terms of two considerations. On the one hand, considering the source nodes and the sink nodes on the connection network as the nests and the food resources, respectively, searching for optimal aircraft routes is quite similar to the foraging behavior of ants. On the other hand, the OAMRP is a heavily constrained problem. As ACO is a constructive meta-heuristic, all constraints can be easily satisfied during the solution construction phase, making coding of ACO for the OAMRP much more straightforward. As a result, ACO algorithms have been widely and successfully applied in various variants of the AMRP [6, 7, 32, 33]. Despite the efficiency of ACO, applying the traditional ACO in the OAMRP with cruise speed control is insufficient and unwise. The conventional ACO selects the next covered flight leg only utilizing the attractiveness of flight connections, while ignoring the information regarding the individual flight leg (i.e., flexible cruise times), which fails to optimize the cruise times. Moreover, the performance of the conventional ACO decreases dramatically in tackling large-scale problems. To model the flexible cruise time, each flight leg is allocated a cruise time window, within which multiple copies of the leg are positioned at a discretized interval. Flight leg copies and thus additional flight connections result in an explosion in the problem size, which poses greater challenge for the conventional ACO.

Therefore, we develop an effective solution method including a preprocessing step for reducing the network size and the improved ACO (IACO) approach to tackle the problem. Computational experiments are performed to demonstrate the efficiency of the proposed solution method.

1.2. Main Contributions and Results. The following is a summary of the primary contributions and results:

- (1) A preprocessing step is designed to decrease the exploded size of the network caused by flexible cruise times.
- (2) A new ACO algorithm is applied to the domain of the OAMRP, which can tackle the proposed problem efficiently. The improvements over the traditional ACO algorithm are described as follows:
 - (i) Firstly, a new state transition strategy is adopted. In addition to the heuristic information on arcs, the heuristic information associated with the flight connection opportunities on nodes is incorporated into the state transition rule. Accordingly, the priority will be given to the nodes with improved flight connection opportunities, allowing for maximum utilization of the cruise time compression.
 - (ii) Secondly, a novel global pheromone updating mechanism is applied. Three new multipliers indicating three new strategies are applied to the term of pheromone increments so that the efficient communication among ants can be achieved. The first strategy, inspired by the study of Naimi and Taherinejad [34], is used to decrease the pheromone increments from the start point to the ending point of a tour, as the freedom of flights selection is gradually restricted during the process of the route construction. Then, to better balance the trade-off between the exploration and the exploitation, the pheromone increments in each iteration are associated with current number of iterations performed, which is achieved by the second strategy. At last, to provide more precise directive information for subsequent searches, we distinguish the contribution of each route to the solution. Hence, the third strategy gives more pheromone increments on the arcs in the route covering more flight legs.
 - (iii) Finally, the pheromone structure and the heuristic function are modified to be more problem-specific. Firstly, two different pheromone structures are set for maintenance arcs and nonmaintenance arcs, respectively. Secondly, an objective-oriented heuristic function, which gives the priority to the feasible flight connections brought by cruise time compression, is employed.

By conducting the experiments using the data from BTS, we first show that the preprocessing step decreases the amount of nodes on the connection network by approximately 70%, thus proving its advantage. Then, the superior performance of the IACO algorithm is testified. It can quickly provide the exact solutions for small-size test cases and reach the best upper bound provided by the CPLEX solver for medium and large-size test cases. Moreover, it remarkably surpasses the existing promising meta-heuristic approaches in terms of solution quality, i.e., the conventional ACO, simulated annealing (SA), and genetic algorithm (GA), while the computation time is slightly longer in medium and large-size test cases, which is acceptable, when compared to the conventional ACO and SA.

1.3. Structure of This Study. Section 2 reviews the literature regarding the OAMRP and cruise speed control. Section 3 provides the problem statement in detail. Section 4 gives mathematical model formulation. The solution method is presented in Section 5. In Section 6, the data sets, experiments, and computational results are described. Finally, Section 7 discusses the conclusion and future research.

2. Literature Review

Recently, there has been a rich set of literature on the OAMRP. In this section, we provide a summary of the most related works in respect to the objective and the solution method.

In terms of the objective, many researches have been conducted on the AMRP with the intention of lowering the number of used aircraft or unutilized flight hours, which can significantly enhance aircraft utilization. Sarac et al. [35] investigated the daily OAMRP, considering the goal of reducing the overall daily maintenance costs. Because quantifying the daily maintenance costs is challenging, the reduction of the total unutilized flying time was chosen as a surrogate objective. This objective was also investigated in the studies of Başdere and Bilge [4] and Al-Thani et al. [5]. Başdere and Bilge [4] discriminated between the critical and noncritical aircrafts to reduce the amount of decision variables in the model formulation. Al-Thani et al. [5] constructed a weighted directed graph, in which an aircraft node served as the representation for an aircraft. Thus, a more compact OAMRP model was developed. In addition to the goal of decreasing the unutilized flying hours, Cui et al. [8] optimized the number of used aircraft for the first time in the OAMRP. However, in this study, the fixed flying time during route construction reduced the options for flight connections, thus affecting the improvement in aircraft utilization. To overcome this limitation, in the study of Zhang et al. [17], they adopted the strategy of cruise speed control during the construction of aircraft routes. Computational experiments demonstrate that aircraft utilization was improved significantly at an expense of a slight increase in the fuel-burn related costs. However, the ACO-based algorithm with limited improvement over the traditional ACO was employed, leading to lengthy computational times and inferior results.

Regarding the solution method, the ACO algorithm has been adopted by the growing literature to efficiently deal with the OAMRP. Eltoukhy et al. [6] studied the OAMRP with flight delay consideration (OAMRPFDD). To represent the nonpropagated delay more explicitly, this study considered several potential scenarios rather than relying on the expected value. Hence, a novel scenario-based stochastic framework for the OAMRPFDD was developed. Although considering various scenarios greatly enlarged the problem size, the ACO algorithm could effectively solve the proposed model. Eltoukhy et al. [33] developed a bilevel nested ACO algorithm so that a Stackelberg game bilevel optimization model could be solved. One specific ACO dealt with the individual level. The dependence between the two levels of the ACO was captured by the feedback of decision variables between two levels. The results proved the efficiency of the proposed ACO algorithm. Eltoukhy et al. [32] incorporated the turnaround time reduction approach into the AMRP model to achieve robustness, and the ACO algorithm was deployed to solve it. Bulbul and Kasimbeyli [9] defined the big-cycle maintenance routing problem, which was solved by the hybrid algorithm combining Gasimov's modified subgradient algorithm and the ACO algorithm. Herein, the ACO was adopted to tackle the subproblem, which minimized the sharp augmented Lagrangian function. They found that ACO is not only simple to implement but also easy to incorporate with other algorithms.

3. Problem Formulation

In this section, the OAMRP with cruise speed control is described first, followed by the presentation of a modified connection network that serves as the foundation for the model formulation and finally the formulation of a novel MILP model for the proposed problem. All notations used are summarized as follows for ease of reference.

Sets

$i, j, l \in L$: set of flight legs

$k \in K_i, n \in K_j, v \in K_l$: set of copies for flight leg i, j , and l , respectively

$t \in T$: set of aircraft

$m \in M$: set of maintenance stations

$c \in \{1, \dots, C\}$: number of maintenance checks

$a \in A$: set of airports

Parameters

FT_{ik} : flying time of the k th copy of flight leg i .

crs_{ik} : cruise time of the k th copy of flight leg i .

S_{ijt}^{kn} : idle time between the k th copy of flight leg i and the n th copy of flight leg j successively operated by an identical aircraft t .

FT_{\max} : maximum allowable flying hours since the last maintenance check.

TO_{\max} : maximum allowable number of takeoffs since the last maintenance check.

C_{daily} : aircraft daily usage cost.

C_{idle} : unit idle time cost.
 d: planning horizon.
 AC: the aircraft type.

3.1. Problem Description. Given a set of flight legs L and a set of aircraft of an identical type T , our study tackles the aircraft routing problem on a four-day basis, considering the maintenance requirements and the flexible flight flying times. In the problem, the reason for adopting a four-day planning horizon is two-fold: (i) it is more practical than a daily horizon, as it allows for daily flight schedule variations to accommodate fluctuating passenger demand and (ii) it is less sensitive to disturbances than a weekly horizon [7]. The maintenance requirements are mandated by the Federal Aviation Administration (FAA) and specify that each aircraft must obtain an A-check at a maintenance station before its cumulative flying hours, cumulative number of takeoffs, or cumulative number of flying days reach the maximum values. In contrast to the traditional OAMRP study, each flight leg's cruise time is allowed to be adjusted within a specified range using cruise speed control. A typical flight flying time is made up of two parts: cruise and noncruise time. In this regard, flexible flight flying time is taken into account for each flight leg.

The objective of this study is to create aircraft routes with high usage of aircraft and thus fewer required aircraft using cruise speed control. The rationale behind is interpreted as follows: For two flights, which can be covered sequentially by the identical aircraft, the latter one's departure time and the former one's arrival time must be separated by enough time. Therefore, there is a lower bound for the departure time of the latter flight. However, this lower bound can be adjusted by changing the former flight's arrival time, which can be accomplished by cruise speed control. Thus, some infeasible flight connections can be changed into feasible ones, allowing more new routes with higher aircraft utilization to be generated.

3.1.1. Numerical Example. This section provides a numerical example to demonstrate the problem definition and the model mechanics. First, we present a flight schedule comprised of 12 flight legs in Table 1 and the turnaround times for involved airports in Table 2. Each flight leg has six features, including departure airport, destination airport, departure time, arrival time, flying time, and cruise time. To cover this flight schedule, four aircrafts are required, as seen in Figure 1 where the routes of different aircrafts are depicted by lines of different colours. When incorporating cruise speed control, it is assumed that the noncruise time is 20 minutes and the cruise time can be reduced to a maximum of 85% of the planned values. In this context, it is intuitive to observe that the required number of aircraft decreases from 4 to 3, which is shown in Figure 2, and

correspondingly, the changes in cruise time, arrival time, and flying time of each flight leg are displayed in Table 3. Accordingly, with this example, the positive impact of flexible cruise times on aircraft utilization improvement has been illustrated.

3.2. Network Structure. In the field of aircraft routing, the connection network structure has been widely used [35–37]. Based on the traditional connection network, we propose a modified connection network to present the AMRP incorporating cruise speed control. This connection network, shown in Figure 3, is comprised of two main elements, i.e., the node set and the arc set, which are created as follows:

The node set consists of the following:

- (i) $\{o, s\}$ where o and s represent the dummy source node and the dummy sink node, respectively.
- (ii) $\{i1, \dots, ik, \dots, j1, \dots, jn, \dots\}$ where node ik represents the k th copy ($k \in K_i$) of flight leg i ($i \in L$). Referring to Marla et al. [38], the flexible cruise time is modelled as below. First, a cruise time window is assigned to each flight leg i defining the range within which its cruise time can be shifted. Then, several copies are placed at a predetermined interval within this time window. Each copy represents a cruise time option, and only one copy is allowed to be chosen for each flight leg during route construction.

Regarding the arc set, it includes the following:

- (i) Flight connection arcs (ik, jn) which are designed for an aircraft to begin a route connect two flight leg copies successively or terminate a route.
- (ii) Maintenance arcs (jn, lv) which are used to schedule a maintenance operation between two successive flight leg copies.

From the modified connection network, it can be observed that modelling flexible cruise times brings a large number of additional nodes and arcs, thereby significantly enlarging the network size.

3.3. Model Formulation. Using the aforementioned notations and the modified connection network as a basis, we define the following decision variables and mathematical model:

Decision Variables

σ_{ik} $\sigma_{ik} = 1$ if the k th copy is chosen for the flight leg i and otherwise $\sigma_{ik} = 0$.

$x_{ijtc}^{kn} \in \{0, 1\}$ $x_{ijtc}^{kn} = 1$ if the aircraft t operates the k th copy of flight leg i and the n th copy of flight leg j successively before receiving the c th maintenance check, and otherwise $x_{ijtc}^{kn} = 0$.

$y_{ijtc}^{kn} \in \{0, 1\}$ $y_{ijtc}^{kn} = 1$ if the aircraft t operates the k th copy of flight leg i and the n th copy of flight leg j

TABLE 1: A flight schedule.

Flight	From	To	Departure time	Arrival time	Flying time	Cruise time
f_1	PHL	MSP	08:30	10:31	121	101
f_2	MSP	PHL	12:20	15:56	216	196
f_3	PHL	IAH	06:00	08:59	179	159
f_4	IAH	PHL	10:30	14:56	266	246
f_5	PHL	CMH	16:00	17:37	97	77
f_6	PHL	DCA	15:10	16:18	68	48
f_7	DCA	CLT	18:30	20:16	106	86
f_8	PHL	CVG	07:00	08:53	113	93
f_9	CVG	PHL	13:31	15:20	109	89
f_{10}	PHL	BUF	16:20	17:46	86	66

TABLE 2: Turnaround time of airports.

Airport	Turnaround time
PHL	41
MSP	51
IAH	50
DCA	46
CVG	40

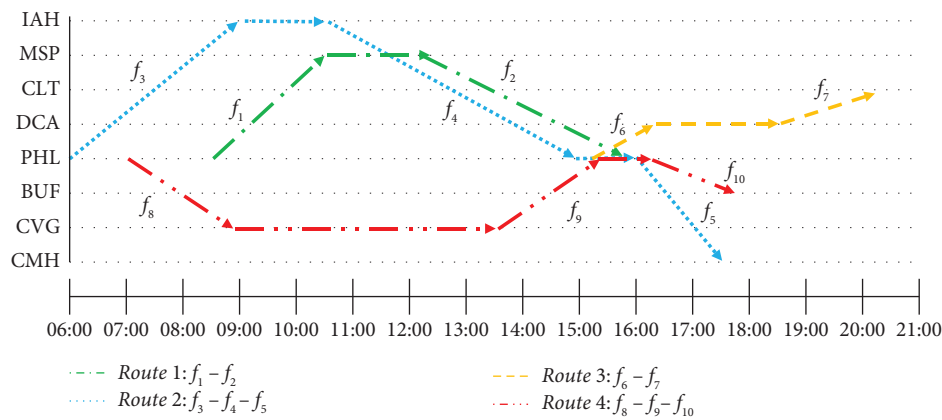


FIGURE 1: The aircraft routing schedule without considering cruise speed control.

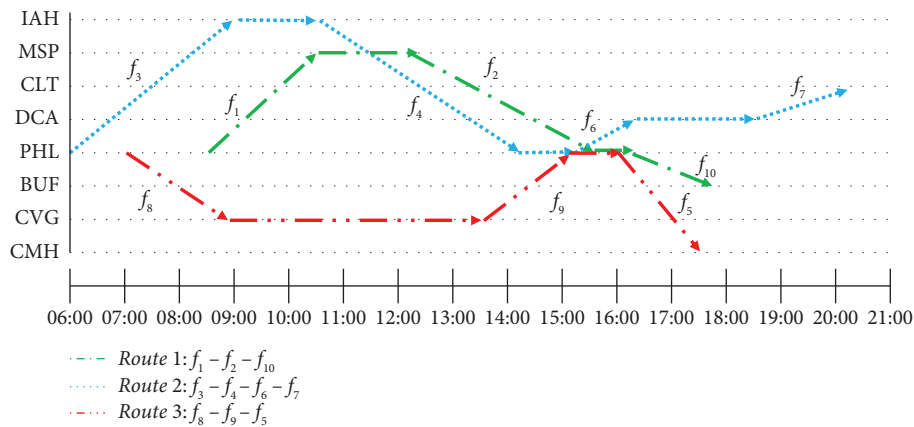


FIGURE 2: The aircraft routing schedule after incorporating cruise speed control.

TABLE 3: The comparison of flight features with and without cruise speed control.

Flight	Arrival time	Flying time	Cruise time	Incorporating cruise speed control		
				Arrival time	Flying time	Cruise time
f_1	10:31	121	101	10:31	121	101
f_2	15:56	216	196	15:31	191	171
f_3	08:59	179	159	08:59	179	159
f_4	14:56	266	246	14:21	231	211
f_5	17:37	97	77	17:37	97	77
f_6	16:18	68	48	16:18	68	48
f_7	20:16	106	86	20:16	106	86
f_8	08:53	113	93	08:53	113	93
f_9	15:20	109	89	15:11	100	80
f_{10}	17:46	86	66	17:46	86	66

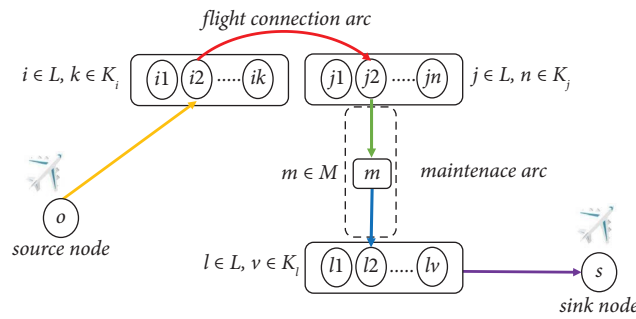


FIGURE 3: The modified connection network.

successively and receives the c th maintenance check in between, and otherwise $y_{ijtc}^{kn} = 0$.

$$\min \sum_{j \in L} \sum_{n \in N_j} \sum_{t \in T} \sum_{c \in C} C_{\text{daily}}(x_{ojtc}^n) d + \sum_{i \in L \cup \{o\}} \sum_{k \in K_i} \sum_{j \in L \cup \{s\}} \sum_{n \in N_j} \sum_{t \in T} \sum_{c \in C} S_{ijt}^{kn} (x_{ijtc}^{kn} + y_{ijtc}^{kn}) C_{\text{idle}} + \sum_{i \in L} \sum_{k \in K_i} \sigma_{ik} F_{ik}^{AC} (crs_{ik}) \quad (1)$$

$$\sum_{k \in K_i} \sigma_{ik} = 1, \quad \forall i \in L, \quad (2)$$

$$\sum_{j \in L} \sum_{n \in N_j} \sum_{i \in L} \sum_{k \in K_i} FT_{ik} x_{ijtc}^{kn} + \sum_{j \in L} \sum_{n \in N_j} \sum_{i \in L} \sum_{k \in K_i} FT_{ik} y_{ijtc}^{kn} \leq FT_{\max}, \quad \forall t \in T, c = 1, \quad (3)$$

$$\sum_{j \in L} \sum_{n \in N_j} \sum_{i \in L} \sum_{k \in K_i} FT_{jn} y_{ijtc}^{kn} + \sum_{j \in L} \sum_{n \in N_j} \sum_{i \in L} \sum_{k \in K_i} FT_{jn} x_{ijtc+1}^{kn} \leq FT_{\max}, \quad \forall t \in T, \forall c \in C, \quad (4)$$

$$\sum_{j \in L} \sum_{n \in N_j} \sum_{i \in L} \sum_{k \in K_i} y_{ijtc}^{kn} + \sum_{j \in L} \sum_{n \in N_j} \sum_{i \in L} \sum_{k \in K_i} x_{ijtc}^{kn} \leq TO_{\max}, \quad \forall t \in T, c = 1, \quad (5)$$

$$\sum_{j \in L} \sum_{n \in N_j} \sum_{i \in L} \sum_{k \in K_i} y_{ijtc}^{kn} + \sum_{j \in L} \sum_{n \in N_j} \sum_{i \in L} \sum_{k \in K_i} x_{ijtc+1}^{kn} \leq TO_{\max}, \quad \forall t \in T, \forall c \in C, \quad (6)$$

$$\sum_{i \in L} \sum_{k \in K_i} \sum_{j \in L} \sum_{n \in N_j} \sum_{c \in C} y_{ijtc}^{kn} \geq 1, \quad \forall t \in T, \quad (7)$$

$$\sigma_{ik} \in \{0, 1\}, \quad \forall i \in L, \forall k \in K_i, \quad (8)$$

$$x_{ijtc}^{kn}, y_{ijtc}^{kn} \in \{0, 1\}, \quad \forall i, j \in L, \forall k \in K_i, \forall n \in K_j, \forall t \in T, \forall c \in C. \quad (9)$$

In the model, we identify the following: (i) the cruise time for each flight leg, (ii) the allocation of the available aircraft to cover all the flight legs, and (iii) the maintenance arrangement for each aircraft. As a result, three decision variables, i.e., σ_{ik} , x_{ijtc}^{kn} , and y_{ijtc}^{kn} , representing the nodes, flight connection arcs, and maintenance arcs, respectively, are defined.

3.3.1. Objective Function. The purpose of this study is to generate aircraft routes with increased usage of the aircraft by considering the flexible cruise times during routes construction. However, additional fuel consumption expenses will be induced. Therefore, a critical trade-off between the increased usage of aircraft and the extra fuel consumption expenses should be made. To quantify aircraft utilization, we consider aircraft usage costs and idle time costs, as increased aircraft utilization results from less idle times of aircraft and fewer aircraft required to cover full range of flight legs. When an aircraft flies two flight legs consecutively, there exists a ground time in between the arrival of the first flight leg and the departure of the second flight leg. This ground time consists of a turnaround time and an idle time. The turnaround time is defined as the necessary time for a landed aircraft to be prepared for its next flight leg. The idle time of an aircraft is the time it spends waiting to operate the next flight leg, which is a waste of a valuable resource. To calculate the idle time costs, we multiply idle times with the unit idle time cost. Aircraft usage costs refer to the expenses induced when the corresponding aircraft is used, which include the fixed operating costs and the lost opportunity cost [39]. As a result, the objective function, explained by equation (1), includes three terms: aircraft usage costs (the first term), idle time costs (the second term), and fuel consumption expenses (the third term). Herein, how to estimate the fuel consumption expenses (i.e., $F_{ik}^{AC}(\text{crs}_{ik})$) is described as follows:

(1) Fuel Consumption Expenses. The influence of cruise time on fuel consumption is investigated by employing the fuel flow model of the cruise stage developed by EUROCONTROL [40]. For the k th copy of flight leg i operated by an aircraft t of type AC , considering its cruise time to be CT_{ik} , the fuel consumption in kg can be calculated as follows:

$$F_{ik}^{AC}(\text{crs}_{ik}) = c_1^{ik,AC} \left(\frac{1}{\text{crs}_{ik}} \right) + c_2^{ik,AC} \left(\frac{1}{(\text{crs}_{ik})^2} \right) + c_3^{ik,AC} (\text{crs}_{ik})^3 + c_4^{ik,AC} (\text{crs}_{ik})^2, \quad (10)$$

where $c_1^{ik,AC} = (1/2) \cdot C_{f_1} \cdot C_{f_{cr}} \cdot C_{D0,CR} \cdot \rho \cdot ws \cdot cd_{ik}^2$, $c_2^{ik,AC} = (1/2) \cdot C_{f_1} \cdot C_{f_{cr}} \cdot ((C_{D0,CR} \cdot \rho \cdot ws \cdot cd_{ik}^3) / C_{f_2})$, $c_3^{ik,AC} = ((1/2) \cdot C_{f_1} \cdot C_{f_{cr}} \cdot (4 \cdot C_{D2,CR} \cdot ma^2 \cdot g^2) / \rho \cdot ws \cdot \cos(\phi)^2 \cdot cd_{ik}^2)$, and $c_4^{ik,AC} = (1/2) \cdot C_{f_1} \cdot C_{f_{cr}} \cdot (4 \cdot C_{D2,CR} \cdot (ma^2 \cdot g^2) / C_{f_2} \cdot \rho \cdot ws \cdot \cos(\phi)^2 \cdot cd_{ik}^2)$. Herein, C_{f_1} , C_{f_2} ,

$C_{f_{cr}}$, $C_{D0,CR}$, and $C_{D2,CR}$ are the fuel consumption coefficients from the Base of Aircraft Data (BADA) user manual [40]. ρ , g , ϕ , ws , ma , and cd_{ik} refer to the air density of the given altitude (kg/m^3), gravitational acceleration (m/s^2), bank angle (degree), wing surface area (m^2), mass of the aircraft (kg), and the distance (m) flown at the cruise stage. According to EUROCONTROL [41], 3.15 kilograms of CO_2 can be emitted by one kilogram of fuel burnt. Consequently, fuel consumption expenses can be defined as follows:

$$F_{ik}^{AC}(\text{crs}_{ik}) = (C_{fuel} + \varepsilon C_{co_2}) F_{ik}^{AC}(\text{crs}_{ik}), \quad (11)$$

where C_{fuel} is the unit fuel price, C_{co_2} is the unit CO_2 emission cost, and ε is the constant associated with CO_2 emission.

3.3.2. Constraints. The constraints defined in this model involve the traditional AMRP constraints and the problem-specific constraints. To highlight the novel features of the proposed problem, their relevant constraints are discussed as follows, while the details of traditional AMRP constraints are provided in the supplementary material.

To model the flexible cruise times, several copies, each with a different cruise time, are allocated to a flight leg. For each flight leg, only one copy is allowed to be selected so that its cruise time can be determined. This requirement is defined by the constraint (2). Besides, constraints (3)–(7) define the three maintenance requirements considered in this study. They denote that an aircraft must receive a maintenance check prior to (i) exceeding the maximal flying time (constraints (3) and (4)), (ii) exceeding the maximal number of takeoffs (constraints (5) and (6)), and (iii) exceeding the maximal number of days (constraint (7)). Herein, it is worth noting that constraint (7) ensures that an aircraft must receive at a minimum of one maintenance check during the planning horizon, i.e., 4-day planning horizon; thus the maintenance requirement of one visit every four days, i.e., maximal number of days, is satisfied. For each aircraft, the accumulated amount of flying time, takeoffs, and flying days are assumed to be 0 at the beginning of the planning horizon.

At last, we define the decision variables' domain in constraints (8) and (9).

4. Solution Method

The network size becomes enormous after modelling the flexible cruise time. Section 5.1 introduces a preprocessing step to simplify the network's structure, and then Section 5.2 presents the IACO algorithm to tackle the presented model.

4.1. Preprocessing. The preprocessing step simplifies the network's structure by removing the redundant nodes. In our study, the flight connection opportunities are improved at the expense of more fuel consumption expenses incurred.

However, for the same flight leg, the nodes incurring more fuel consumption expenses may not necessarily bring more flight connection opportunities. The preprocessing step focuses on these nodes. To be specific, nodes of the same flight leg are compared from two aspects: the number of connectable nodes, denoting its flight connection opportunities, and the degree of cruise time compression, denoting the extra fuel consumption expenses incurred. Among all nodes of the same flight leg, the nodes that cannot increase the number of connectable nodes and have larger degree of cruise time compression will be removed. This preprocessing step can achieve large network size reduction without affecting the connection network's characteristics, particularly when copies are placed in a finer interval.

4.2. An IACO Algorithm. The ACO, as originally designed by Dorigo, was encouraged by the behaviour of real ant colonies searching for food. An ant deposits one special type of secretion, called pheromone, on the way from their nest to food source, and in turn this pheromone trail is used to guide the successor ants. By this indirect but useful communication form, positive feedback can be achieved, and thereby the shortest path between the nest and the food resource can be found. In the ACO algorithm, artificial ants mimic this foraging behaviour. They walk on a weighted and constrained graph, over which the combinatorial optimization problem is modelled. Then, they identify the optimal paths based on the pheromone trails that they laid to record the promising searching space and the problem-dependent heuristic information.

In our study, an IACO algorithm is developed. Next, we will explain the main elements of the proposed algorithm. The notations required for the IACO are defined as follows:

- Attr_{ikjn}: the attraction of arc (ik, jn) for an ant
- q_0 : a predefined parameter ($0 < q_0 < 1$)
- q : a random number (which is uniformly distributed in the range of 0 to 1)
- ρ : the parameter that controls the evaporation rate ($0 < \rho < 1$)
- τ_{ikjn} : the pheromone trail deposited on arc (ik, jn)
- η_{ikjn} : the heuristic information on arc (ik, jn)
- φ_{jn} : the heuristic information on node jn
- α, β, γ : the parameter to adjust the relative influence of τ_{ikjn}, η_{ikjn} , and φ_{jn}
- $\Delta\tau_{ikjn}^{\text{best}}$: the pheromone increments on arc (ik, jn) which is contained in the current best-found solution at each iteration
- N_{ik} : set of feasible nodes which can be connected with node ik .

4.2.1. State Transition Mechanism. When an ant is on the node ik , the following state transition mechanism will be utilized to select the next node jn :

$$f(x) = \begin{cases} \arg\text{-max}_{jn \in N_{ik}} \{ \text{Attr}_{ikjn} \}, & \text{if } q \leq q_0, \\ P_{ikjn}, & \text{if } q > q_0. \end{cases} \quad (12)$$

When q is less than the predefined parameter q_0 , the ant will choose the node jn with the biggest Attr_{ikjn} as the next node. Otherwise, the ant will select the next node based on the probability P_{ikjn} defined by the formula, which is as follows:

$$P_{ikjn} = \frac{\text{Attr}_{ikjn}}{\sum_{dw \in N_{ik}} \text{Attr}_{ikdw}} \text{ if } jn \in N_{ik}. \quad (13)$$

And the attraction value of arc (ik, jn) for an ant is as follows:

$$\text{Attr}_{ikjn} = [\tau_{ikjn}]^\alpha [\eta_{ikjn}]^\beta [\varphi_{jn}]^\gamma. \quad (14)$$

In the conventional ACO algorithm, an ant incrementally constructs solution on the graph, where all the nodes must be visited at least once and only once. Accordingly, only the attractiveness of arcs is considered in the state transition mechanism and the sequencing is the only issue to be concerned. However, in our study, due to the existence of the flexible cruise time, each flight leg has several leg copies and only one copy is allowed to be chosen, implying that not all the nodes on the modified connection network are to be covered. The node selection is also crucial. Therefore, the heuristic information of the node jn (φ_{jn}) is introduced as a guide for node selection. Subsequently, we will address the τ_{ikjn} , η_{ikjn} , and φ_{jn} in detail.

(1) *Pheromone Information of Arc (ik, jn) :* τ_{ikjn} . The pheromone information represents the ants' search experience and directs the future ants to search in the promising space. Traditionally, only one pheromone structure is involved as all arcs are of the same character. However, the proposed connection network is made up of two distinct types of arcs including nonmaintenance arcs and maintenance arcs so as to comply with mandatory maintenance requirements. Consequently, two pheromone structures are set for two types of arcs, thus allowing for more accurate representation of search experience.

(2) *Heuristic Information of Arc (ik, jn) :* η_{ikjn} . Typically, the heuristic information is based on the problem's characteristic. The arcs with less idle time are favoured because the objective of this study is to minimize the quantity of necessary aircraft, and as a result, they have a higher heuristic value. Therefore, the heuristic value is inversely proportional to the value of idle time. Moreover, when an ant is positioned on the node with cruise time compression, it is reasonable to give the priority to the arcs that are only feasible after this cruise time compression. In the new heuristic function, these arcs are distinguished from the originally feasible arcs and their heuristic values are doubled. The addition of one is utilized to avoid division by 0. The heuristic function is defined as follows:

$$\eta_{ikjn} = \begin{cases} \frac{1}{\left(\sqrt{S_{ijt}^{kn}} + 1\right)}, & \text{if arc } (ik, jn) \text{ is the originally connected arcs,} \\ \frac{2}{\left(\sqrt{S_{ijt}^{kn}} + 1\right)}, & \text{otherwise.} \end{cases} \quad (15)$$

(3) *Heuristic Information of Node jn : φ_{jn} .* In the list of feasible nodes to be next covered, the nodes of the same flight leg involve the identical heuristic information of arc (ik, jn) . However, they have different cruise times, resulting in different flight connection opportunities that have direct impact on the route construction. The node with improved flight connection opportunities can bring more efficient flight connections and thus be more preferred. Therefore, the heuristic information of node jn (φ_{jn} , eq. (16)) is introduced.

$$\varphi_{jn} = \frac{FCO_{jn}}{FCO_{jr}}, \quad (16)$$

where the FCO_{jn} and the FCO_{jr} denote the number of nodes that node jn and node jr can connect, respectively. Node jr represents the r th copy of flight leg j that does not have cruise time compression.

4.2.2. *Global Pheromone Trail Update.* After each iteration, the pheromone trail on the arcs is updated using the following equation:

$$\tau_{ikjn} \leftarrow (1 - \rho)\tau_{ikjn} + \Delta\tau_{ikjn}^{\text{best}}, \quad (17)$$

where the first term reduces the amount of pheromone on all the arcs and the second term gives the pheromone increments on the arcs of the current best-found solution. The pheromone increments are calculated using the formula which is as follows:

$$\Delta\tau_{ikjn}^{\text{best}} = A \times B \times C \times \left(\frac{Q}{\text{cost}(R_{\text{best}})}\right), \quad (18)$$

$$A = e^{-5((nr+nd)/(NR+ND))}, \quad (19)$$

$$B = \frac{1}{(1 + 0.95^{it})}, \quad (20)$$

$$C = ND. \quad (21)$$

In the conventional ACO, the pheromone increments on the arcs included in the current best-found solution are set to be $Q/\text{cost}(R_{\text{best}})$. $\text{cost}(R_{\text{best}})$ is the best objective value found from the start till the present iteration. Q is the control parameter that is used to adjust the amount of deposited pheromone and thus helps the algorithm avoid local optimal convergence or excessively random search. To increase search efficiency and precision, three multipliers A , B , and C are applied to the term of pheromone increments. In the following, we will introduce these three multipliers in detail.

(1) *Multiplier A.* As was mentioned and proved in the study of Naimi and Taherinejad [34], giving all the arcs the same pheromone increments is not a good strategy for pheromone updating. At the initial stage of route construction, an ant has greater freedom in choosing the next flight leg to be covered since few flight legs have been covered and are prohibited to be chosen. In contrast, at the final stage, most of the flight legs have been covered which strongly constrains the available flight legs. Accordingly, a poor decision made at the initial stage may lead to a series of poor choices and, eventually, a disastrous solution. Therefore, it is rational to allow up-to-now best ant to have greater impact on pheromone updating at its initial stage of a tour and less impact when it is about to finish its tour. Based on above discussion, the multiplier A is proposed. In this eq. (20), NR is the number of routes in the solution, nr denotes the nr th route that the arc (ik, jn) is in, ND is the amount of nodes covered in nr th route, and nd signifies the amount of nodes covered before choosing arc (ik, jn) in nr th route. Apparently, both nr and nd increase from the beginning to the ending of a solution. Therefore, the multiplier A decreases towards zero ($e^{-5} \approx 0$), accurately reflecting the gradually diminishing role of the up-to-now best ant from the beginning to the ending of a tour in the pheromone updating.

(2) *Multiplier B.* During the process of iterations, the solutions' quality will be greatly improved, and the ants will gradually shift their focus from the exploration to the exploitation. Therefore, fewer pheromone increments should be applied to the earlier iterations to increase the solutions' diversity and avoid a rapid fall into the local optimal, while more pheromone increments should be applied to the later iterations to strengthen the exploitation and accelerate the convergence. To achieve this purpose, multiplier B (eq. (20)) is introduced, where it denotes the iteration index. This multiplier increases the pheromone increments during the process of iterations, allowing for dynamically balancing the exploration and the exploitation so as to improve the search efficiency.

(3) *Multiplier C.* The routes covering more flight legs are more favourable due to its higher aircraft utilization. Thus, it is logical to allocate more pheromone increments to the route covering more flight legs, which enables the subsequent ants to perform more delicate searches in a more favourable area in the next iteration. Therefore, we propose the multiplier C (eq. (20)), which denotes the total amount of nodes covered by the nr th route which contains the arc (ik, jn) .

4.2.3. *The Proposed Algorithm.* The algorithm Pseudocode 1 is summarized as follows:

In this algorithm, the mechanism of the global pheromone trail update is defined in Section 4.2.2 while the Pseudocode 2 of route construction is described as follows:

5. Computational Experiment

The computational study is carried out in this section with two objectives: (i) to evaluate the efficiency of the preprocessing step and (ii) to validate the effectiveness of the IACO algorithm. All the algorithms were programmed in MATLAB R2016a, whereas the MILP model was programmed in CPLEX 12.1. The experiments were conducted on a Windows 10 laptop with Intel Core i7, 2.6 GHz CPU, and 16 GB RAM.

5.1. *Test Instance and Experimental Setup.* To our best knowledge, there are no standard benchmarks in the literature for the AMRP with cruise speed control. Therefore, we collect ten cases of flight schedules from the BTS [42] database. Table 4 presents the main characteristics of the collected test cases involving small-size, medium-size, and large-size cases to demonstrate that the presented algorithm is applicable to real-world problems. Generally, the majority of airlines consider schedules with fewer than 250 flight legs to be small-size cases [7]. Consequently, test cases 1 to 5 and test cases 6 to 10 are classified as small-size cases and medium-size and large-size cases, respectively.

Based on airlines' recommendation, we assume that the maximal flying hours are 40 hours, the maximal number of take-offs are 8, and the duration of the maintenance check is 6 hours for all test cases. According to Duran et al. [43], we assume that the unit CO_2 emission cost is \$0.02/kg, the scheduled noncruise time is 20 minutes, the cruise time window is $[85\% \times \text{original cruise time}, \text{original cruise time}]$, and the copy interval is $5\% \times \text{original cruise time}$. Referring to the IATA fuel price monitor, the unit fuel cost is assumed to be \$0.33/kg. This study considers the most representative aircraft type, A320 111, for which Table 5 provides the specific information. Herein, it is worth noting that different aircrafts of the same type may have different fuel consumption coefficients due to different maintenance history and engine efficiency. However, these differences are hard to quantify. According to Aktürk et al. [44], we assume that aircrafts of the same type have the same fuel consumption coefficients.

Before adopting the IACO algorithm, a parameter tuning process is necessary due to the direct influence of parameters on the performance of the IACO algorithm. By using the Taguchi method, the best parameters are set as $\alpha = 1, \beta = 1, \gamma = 1, q_0 = 0.95, \rho = 0.15$, colony_size = the amount of nodes on the connection network after the preprocessing step, and iter_max = 500. To examine the average performance, the algorithm runs 30 times for each test case since no better result could be obtained by increasing the number of runs.

5.2. *Impact of the Pre-Processing Step.* For the purpose of examining the effectiveness of the preprocessing step in the reducing network size, we compare the amount of nodes on the connection network before and after the preprocessing step. The findings are displayed in Table 6. It is intuitive to observe that the amount of nodes on the connection network is reduced by approximately 70% in all test cases. Additionally, it is of great significance to note that the solution quality is unaffected because only copies that do not improve the flight connection opportunities were deleted, as mentioned in Section 4.1.

To make a fair comparison, all the solution methods are implemented after applying this preprocessing step.

5.3. *Performance Validation of the IACO Algorithm.* This subsection first compares the solutions obtained from the IACO algorithm with that provided by the CPLEX solver. Then, the IACO algorithm is compared with three meta-heuristic approaches, namely the convention ACO, the simulated annealing (SA), and the genetic algorithm (GA). The comparison with the conventional ACO is necessary so as to examine the efficacy of the improvements while the GA and SA are adopted as the comparison algorithms due to their demonstrated effectiveness in coping with the AMRP [7].

5.3.1. *Performance Comparison of the IACO with the CPLEX Solver.* In this part, the exact solutions and the upper bounds derived by the CPLEX solver are utilized as criteria for the small-size test cases and the medium-size and large-size test cases, respectively, because it is challenging for the CPLEX solver to optimally deal with the medium-size and large-size problems in an acceptable computing time. Note that the upper bound is achieved within 7 hours running time limitation of the CPLEX solver.

(1) *Small-Size Problem Analysis.* Table 7 displays the comparison between the solutions provided by the CPLEX solver and those obtained from the IACO algorithm. The first column indicates the test case number. The next two columns show the solutions obtained from the CPLEX solver, where R^* column and T column represent the exact solution and the computation time, respectively. In the remaining columns, the solutions derived by the proposed algorithm are presented. R_{best} column, R_{ave} column, σ column, and \bar{T} column describe the best solution, the average solution, the standard deviation, and the average computation time, respectively. The % Gap column reports the percentage difference between the average solution and the optimal solution, which is calculated using the formula (22) which is as follows:

$$\% \text{Gap} = 100 \times \frac{(R_{\text{ave}} - R^*)}{R^*}. \quad (22)$$

From Table 7, it can be observed that the values of both R_{ave} and R_{best} are the same with the value of R^* for the first test case. For the other four test cases, the values of R_{best} still

Input: A set of flight legs, the copies for each flight leg, and a set of aircraft of an identical type
Output: Aircraft routes that cover all the flight legs and the cruise time for each flight leg

- (1) **Initialization**
- (2) Parameters: $\alpha, \beta, \gamma, \rho, q_0, Q$
- (3) iter_max: the maximum number of iterations performed by the algorithm
- (4) Ant colony ANT
- (5) Best solution BS
- (6) **For** iter \leftarrow 1 to iter_max **do**
- (7) **For** $n \leftarrow$ 1 to colony_size **do**
- (8) initiate the solution of ant n $S_n = \emptyset$
- (9) Great list FN storing copies of all flight legs and list T of aircraft
- (10) **While** the list FN is not empty **do**
- (11) Select an aircraft t from the list T
- (12) *Route construction* for aircraft t
- (13) Add the route of aircraft t to solution S_n
- (14) **End while**
- (15) **End for**
- (16) Make an update for the current best solution using solution S_n
- (17) *Global pheromone trail update*
- (18) **End for**
- (19) Return BS

PSEUDOCODE 1: The IACO Algorithm.

- (1) Place the aircraft t on the source node o and start route construction R_t
- (2) Add the k th copy of flight leg i that has the earliest departure time to R_t
- (3) Create the list AF storing the feasible flight leg copies that can be next covered using time and location constraints (10) and (13)
- (4) **While** AF is not empty **do**
- (5) Create the list PF storing the flight leg copies that satisfy the maintenance requirements
- (6) **If** the list PF is not empty **do**
- (7) Choose the next n th copy of flight leg j from the list PF using equation (23) and (24)
- (8) Add the n th copy of flight leg j to R_t
- (9) Update the list AF
- (10) **Else**
- (11) **If** the aircraft is maintenance feasible considering location constraint (12), the working times and the capacity constraints (18)–(20) **do**
- (12) Add a maintenance operation to R_t
- (13) Initialize the accumulative flying hours, the accumulative number of takeoffs, and the accumulative number of flying days
- (14) Create the list RF storing the feasible flight leg copies that can be next covered using time and location constraints (11) and (14)
- (15) **If** the list RF is not empty **do**
- (16) Choose the next e th copy of flight leg l from the list RF using equations (23) and (24)
- (17) Add the e th copy of flight leg l to R_t
- (18) Update the list AF
- (19) **End**
- (20) **End**
- (21) **End**
- (22) **End**
- (23) Terminate R_t at the sink node s

PSEUDOCODE 2: Route construction.

equals to the values of R^* while the values of R_{ave} slightly differ from the values of R^* by no more than 0.42%. From σ column, it shows that there is no variance in the solutions obtained from the IACO algorithm in the first test case while the variance slightly increases with the increasing size of the

last four test cases. These results well demonstrate the accuracy and the robustness of the ICAO algorithm.

Regarding the computation time, the T and \bar{T} column show that the proposed algorithm can achieve the solutions within 70 seconds, whereas the CPLEX solver requires up to

TABLE 4: Test cases.

Test cases	Flight legs	Fleet size	Airports	Maintenance stations
1	40	8	14	7
2	83	11	24	10
3	120	15	38	16
4	150	15	42	16
5	222	21	53	18
6	260	25	52	18
7	312	34	56	18
8	410	45	57	21
9	500	43	54	21
10	613	62	57	21

TABLE 5: Aircraft parameters.

Aircraft type: A320 111	
Mass (kg) 62000	$C_{D0,CR}$ 0.024
Reference wing area (m ²) 122.4	$C_{D2,CR}$ 0.0375
MRC speed (km/h) 855.15	Cf_1 0.94
Base turnaround time (min) 28	Cf_2 50000
Unit idle time cost (\$/min) 136	Cf_{CR} 1.095
Daily usage costs (\$/day) 81600	

TABLE 6: The comparison of the number of nodes on the connection network.

Test cases	Flight legs	Nodes		Percentage decrease (%)
		Before preprocessing	After preprocessing	
1	40	160	46	71.25
2	83	332	96	71.08
3	120	480	133	72.29
4	150	600	164	72.67
5	222	888	238	73.20
6	260	1040	282	72.88
7	312	1248	338	72.92
8	410	1640	461	71.89
9	500	2000	579	71.05
10	613	2452	683	72.15

TABLE 7: Performance comparison with CPLEX for small-size cases.

Test cases	CPLEX		IACO				% Gap
	R^*	T (s)	R_{best}	R_{ave}	σ	\bar{T} (s)	
1	942,453	10.43	942,453	942,453	0	4.58	0
2	5,696,567	78.62	5,696,567	5,702,683	2,031	10.57	0.11
3	12,098,788	302.31	12,098,788	12,128,159	8,728	20.79	0.24
4	5,850,835	472.45	5,850,835	5,863,069	2,936	28.07	0.21
5	8,705,534	10215.46	8,705,534	8,741,893	18,749	64.22	0.42

3 hours. Apparently, it takes much less time for the IACO algorithm in comparison with the CPLEX solver. Moreover, the computation time greatly increases with the increasing

size of the test cases for the CPLEX solver, whereas no obvious variation in the computation time is found for the IACO algorithm.

To summarize, the IACO algorithm can achieve the significant computation time reduction in comparison with the CPLEX solver while still producing good-quality solutions as the best solutions are the same with the optimal solutions and the average solutions differ from the optimal solutions by no more than 0.42%.

(2) *Medium- and Large-Size Problem Analysis.* Table 8 summarizes the comparison results, which contain similar information as Table 7 except for the *UB* column. The *UB* column denotes the upper bound provided by the CPLEX solver within the time limitation.

From Table 8, we can observe that the IACO algorithm still significantly outperforms the CPLEX solver. Regarding the solution quality, the values of R_{best} are identical to the values of *UB* in all test cases, whereas the values of R_{ave} differ from the values of *UB* by no more than 12%. Moving to computational time, albeit the considerably enlarged network, our algorithm can still solve it within 15 minutes, which is reasonable and practical. Moreover, taking a close look at the σ , although the values increase, they still imply the small solution variances as the corresponding solution values increase greatly. This confirms the robustness of the IACO algorithm in dealing with medium-size and large-size problems.

5.3.2. Performance Comparison of the IACO with Three Meta-Heuristic Approaches. In this part, we solve the proposed model using three meta-heuristic approaches, namely the convention ACO, SA, and GA, under the same experimental conditions, and then compare the average performance. The parameters settings are identical for the conventional ACO algorithm and the ICAO algorithm. For GA and SA, we followed the procedures and parameters introduced by Eltoukhy et al. [7], taking into account flexible cruise time. The performance of three meta-heuristics is summarized in Table 9, while the percentage improvement of IACO over the three approaches in solution quality is illustrated in Table 10 and Figures 4 and 5. Herein, IMP% is computed by $(R_{\text{CAL}} - R_{\text{IACO}})/R_{\text{IACO}}$ where R_{IACO} and R_{CAL} denote the solution of the IACO and the solution of the algorithm that is compared with the IACO, respectively. Additionally, the computation time comparisons are displayed in Table 11 and Figures 6–8.

From Table 9, it can be interpreted that the conventional ACO algorithm can produce considerably superior solutions in comparison with SA and GA and even be capable to generate the exact solution for the particularly small-size case such as test case 1. Moreover, its computation time slightly increases in some of test cases as compared to SA but is far less than that of GA for all test cases, as is evident from Table 11. For instance, in test case 10, the computation time of GA is up to 1 hour whereas that of the other two is around 10 minutes. Although SA consumes less time to produce the solutions, the solution quality is the poorest. In case 1, the solution obtained by SA significantly deviates from the optimality, even up to 130.25%.

Moving to the IACO algorithm, we can see from Tables 7 and 8 that the average gap of IACO with the exact solution for small-size test cases is within 0.5%, while that with the upper bound is within 12% for medium-size and large-size test cases, which is fairly less in comparison with the other three approaches, particularly SA and GA. Moreover, its remarkable outperformance in term of the solution quality is visible in Table 10 and Figures 4 and 5. Along with the significant improvement in solution quality, the computation time of the IACO algorithm is slightly longer compared to the conventional ACO and SA for the medium-size and large-size test cases, while in some of small-size test cases, such as case 2, 3, and 4, the computation time is even shorter. This slight increase in computation time is worthy in comparison with the considerable improvement in solution quality. Taking case 7 as an example, 32 seconds increase in computation time can improve the average solution quality by around 39.79% in comparison with SA, while 21 seconds increase in computation time can result in 16.98% improvement in average solution quality in comparison with the conventional ACO.

We will explain the rationalities behind the above results in the following.

In the OAMRP with cruise speed control, different from the traditional AMRP, each flight leg has several leg copies, each copy involves a cruise time option, and only one copy can be selected. In SA, it starts with an initial solution in which the copy for each flight leg has been determined and then improves the current solution by local search in which the copy for each flight leg has been fixed. As a result, the cruise speed control only plays the role in the initial route construction. This significantly diminishes the beneficial effect of cruise speed control on route construction, resulting in great reduction in the solution space. Thus, the solutions obtained by SA are the poorest. In the GA, the solution quality is improved based on a population of solutions. The solution space is enlarged as compared to SA due to more choices of copies for each flight leg, and better solutions can be produced.

However, in the conventional ACO algorithm and the IACO algorithm, they construct a new solution from an empty solution. The copy for each flight leg dynamically changes at each time of route construction, which fully utilizes the flexible cruise time. Therefore, their solutions are far much better as compared to SA and GA. While comparing the conventional ACO algorithm and the IACO algorithm, the solution quality of the IACO algorithm can be greatly improved without significant time variance due to the efficacy of the new strategies.

Based on the above discussions, we can get the conclusion that the IACO algorithm notably outperforms the existing approaches regarding the solution quality while the computation time slightly increases as compared to SA and the conventional ACO in the medium-size and large-size test cases. Hence, we have verified the efficiency of the IACO algorithm.

TABLE 8: Performance comparison with CPLEX for medium-size and large-size cases.

Test cases	CPLEX		IACO			% Gap
	UB	R_{best}	R_{ave}	σ	\bar{T} (s)	
6	18,461,505	18,461,505	19,175,821	171,221	98.52	3.73
7	13,268,536	13,268,536	13,804,348	392,180	167.94	4.04
8	16,030,559	16,030,559	17,573,417	558,061	328.55	9.62
9	33,141,047	33,141,047	34,095,754	740,870	526.82	2.88
10	23,782,212	23,782,212	26,415,509	1,073,948	796.09	11.07

TABLE 9: Performance of the other three approaches.

Test cases	UB	The conventional ACO			SA			GA		
		R_{best}	R_{ave}	% Gap	R_{best}	R_{ave}	% Gap	R_{best}	R_{ave}	% Gap
1	942,453	942,453	942,453	0	2,169,998	2,169,998	130.25	1,653,063	1,653,063	75.40
2	5,696,567	5,925,061	5,931,915	4.13	8,358,922	8,423,818	47.88	8,226,557	8,228,362	44.44
3	12,098,788	12,475,909	12,490,795	3.24	14,003,494	14,166,171	17.09	13,960,910	13,980,505	15.55
4	5,850,835	6,084,835	6,166,776	5.40	9,151,499	9,216,232	57.52	8,462,961	8,489,085	45.09
5	8,705,534	9,540,768	9,557,255	9.78	11,979,579	12,023,937	38.12	11,267,337	11,312,610	29.95
6	18,461,505	20,712,426	20,865,974	13.02	22,120,281	22,448,473	21.60	21,234,926	21,677,603	17.42
7	13,268,536	15,843,229	16,148,909	21.71	19,206,652	19,296,554	45.43	18,474,716	18,893,036	42.39
8	16,030,559	17,750,186	18,610,041	16.09	24,368,613	25,163,323	56.97	22,112,424	23,250,217	45.04
9	33,141,047	37,166,883	37,672,274	13.67	39,372,149	40,677,344	22.74	38,101,892	39,213,505	18.32
10	23,782,212	28,044,151	29,178,249	22.69	31,057,124	35,822,348	50.63	30,030,487	33,571,978	41.16

TABLE 10: Improvement over the other three approaches.

Test cases	Average solution (IMP%)			Best solution (IMP%)		
	The conventional ACO (%)	SA (%)	GA (%)	The conventional ACO (%)	SA (%)	GA (%)
1	0.00	130.25	75.40	0.00	130.25	75.40
2	4.02	47.72	44.29	4.01	46.74	44.41
3	2.99	16.80	15.27	3.12	15.74	15.39
4	5.18	57.19	44.79	4.00	56.41	44.65
5	9.33	37.54	29.41	9.59	37.61	29.43
6	8.81	17.07	13.05	12.19	19.82	15.02
7	16.98	39.79	36.86	19.40	44.75	39.24
8	5.90	43.19	32.30	10.73	52.01	37.94
9	10.49	19.30	15.01	12.15	18.80	14.97
10	10.46	35.61	27.09	17.92	30.59	26.27

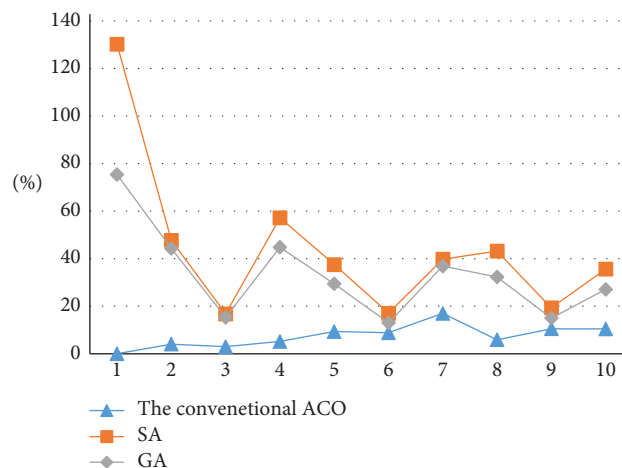


FIGURE 4: Improvement in average solution over the other three approaches.

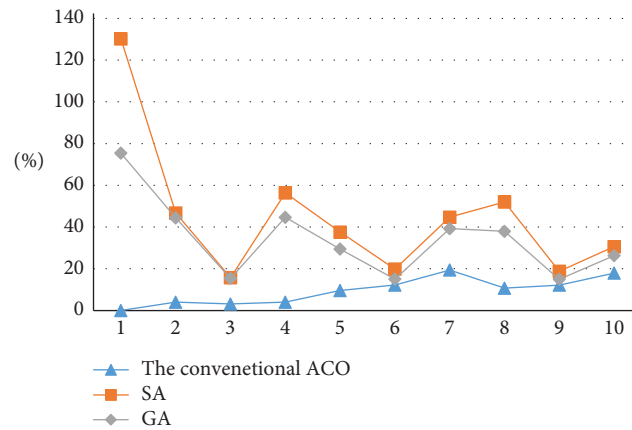


FIGURE 5: Improvement in best solution over the other three approaches.

TABLE 11: Computation time comparison.

Test cases	Computation time (s)			
	The conventional ACO	SA	GA	IACO
1	4.85	3.83	12.12	4.58
2	11.51	12.63	36.54	10.57
3	27.82	29.23	66.13	20.79
4	34.67	32.97	92.64	28.07
5	66.19	64.18	448.98	64.22
6	92.55	88.04	523.74	98.52
7	146.47	135.32	908.98	167.94
8	301.98	296.42	1424.32	328.55
9	464.45	470.62	2917.51	526.82
10	735.76	601.91	3375.29	796.09

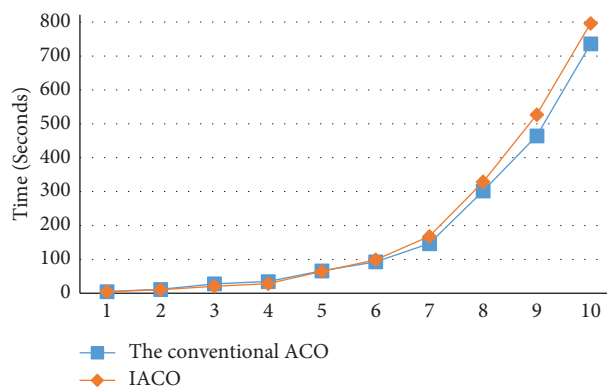


FIGURE 6: The computational time comparison with the conventional ACO.

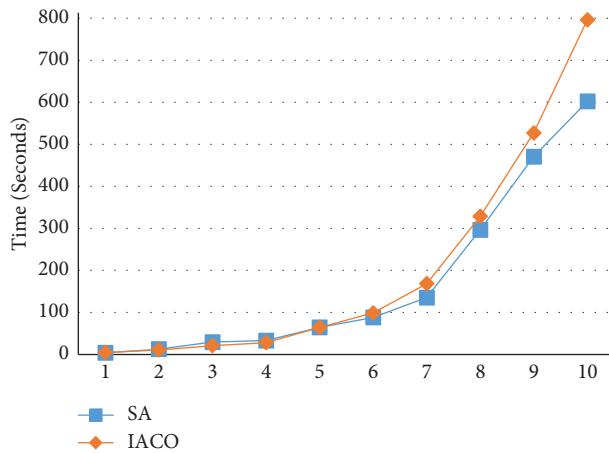


FIGURE 7: The computational time comparison with SA.

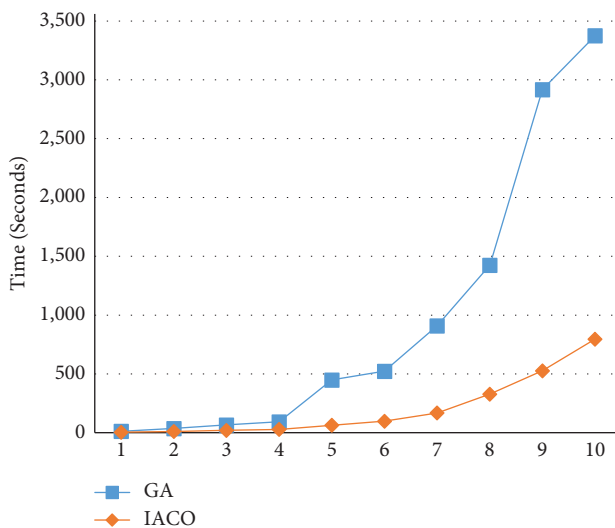


FIGURE 8: The computational time comparison with GA.

6. Conclusions

Due to the growing trend of aircraft operating lease, airlines are challenged by its expensive leasing rental situation. To achieve profitability, it is of great significance to reduce the number of required aircraft. In this study, we investigate a new variation of the OAMRP, namely the OAMRP incorporating cruise speed control, in which the improved aircraft utilization and thus the reduced number of required aircraft can be achieved by changing flying time in aircraft routing [17]. Given its NP-hard nature and the additional complexity added by modeling the flexible cruise time, the challenge lies in how to efficiently solve it. To tackle the problem, this study first designs a preprocessing step to reduce the size of the network. Then, an IACO algorithm is developed by proposing a new state transition strategy and a novel pheromone updating mechanism. In the state transition strategy, both the heuristic values on nodes and arcs are used to compute the set of feasible neighbours. In the novel pheromone updating mechanism, the pheromone increments include three multipliers indicating three new strategies to improve the communication efficiency between ants.

Moreover, the pheromone structure and the heuristic function are modified to be more problem specific.

Based on data sets derived from the BTS, two computational experiments are carried out. Firstly, we examine the effectiveness of the preprocessing step in reducing the network size. The results show that the preprocessing step can reduce the amount of nodes on the connection network by around 70% without compromising the solution quality. Secondly, the performance of the IACO algorithm is verified by comparing with the CPLEX solver and three promising meta-heuristic approaches, including the conventional ACO, SA, and GA. From the results analysis, the IACO algorithm can achieve substantial computation time savings and makes an excellent compromise between the solution quality and the computation time in comparison with the CPLEX. As compared to the traditional ACO, SA, and GA, the IACO algorithm improves the best solutions by 9.31%, 45.27%, and 34.27%, and the average solutions by 7.42%, 44.45%, and 33.35%, while the computation times are slightly longer in some of the test cases.

Despite the contributions made in our study, we did not take into account operational uncertainties such as flight delays. Because flight delays occur frequently in practice, one of our future research directions is to consider flight delays during route construction so that the generated route is less susceptible to disruptions. Moreover, numerous novel ACO variants which have been demonstrated to be highly effective for a variety of optimization purposes emerged recently [18–21, 25–29]. However, they have not been applied in airline schedule planning yet. Employing these new algorithms to solve the proposed problem and comparing them with the proposed IACO algorithm will also be an interesting future direction.

Data Availability

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

Conflicts of Interest

The authors declare that there are no conflicts of interest.

Acknowledgments

The work described in this article was substantially supported by grants from the Macau University of Science and Technology Faculty Research Grants (FRG) under grant number FRG-22-108-MSB, and The Macau Foundation Fund (MFP) under grant number MF-23-008-R.

Supplementary Materials

Traditional AMRP constraints. (*Supplementary Materials*)

References

- [1] Z. H. Zhan, J. Zhang, Y. Li et al., "An efficient ant colony system based on receding horizon control for the aircraft arrival sequencing and scheduling problem," *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, no. 2, pp. 399–412, 2010.

- [2] L. J. Wu, Z. H. Zhan, X. M. Hu, P. Guo, Y. Zhang, and J. Zhang, "Multi-runway aircraft arrival scheduling: a receding horizon control based ant colony system approach," in *Proceedings of the 2019 IEEE Congress on Evolutionary Computation (CEC)*, pp. 538–545, Wellington, New Zealand, June 2019.
- [3] S. Z. Zhou, Z. H. Zhan, Z. G. Chen, S. Kwong, and J. Zhang, "A multi-objective ant colony system algorithm for airline crew rostering problem with fairness and satisfaction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 11, pp. 6784–6798, 2021.
- [4] M. Başdere and Ü. Bilge, "Operational aircraft maintenance routing problem with remaining time consideration," *European Journal of Operational Research*, vol. 235, no. 1, pp. 315–328, 2014.
- [5] N. A. Al-Thani, M. Ben Ahmed, and M. Haouari, "A model and optimization-based heuristic for the operational aircraft maintenance routing problem," *Transportation Research Part C: Emerging Technologies*, vol. 72, pp. 29–44, 2016.
- [6] A. E. Eltoukhy, F. T. Chan, and S. H. Chung, "Scenario-based stochastic framework for operational aircraft maintenance routing problem," in *Proceedings of the Paper Presented at the the International Conference on Systems Engineering and Engineering Management*, Tokyo, Japan, October 2017.
- [7] A. E. Eltoukhy, F. T. Chan, S. H. Chung, B. Niu, and X. P. Wang, "Heuristic approaches for operational aircraft maintenance routing problem with maximum flying hours and man-power availability considerations," *Industrial Management and Data Systems*, vol. 117, no. 10, pp. 2142–2170, 2017.
- [8] R. Cui, X. Dong, and Y. Lin, "Models for aircraft maintenance routing problem with consideration of remaining time and robustness," *Computers and Industrial Engineering*, vol. 137, Article ID 106045, 2019.
- [9] K. G. Bulbul and R. Kasimbeyli, "Augmented Lagrangian based hybrid subgradient method for solving aircraft maintenance routing problem," *Computers and Operations Research*, vol. 132, Article ID 105294, 2021.
- [10] T. H. Oum, A. Zhang, and Y. Zhang, "Optimal demand for operating lease of aircraft," *Transportation Research Part B: Methodological*, vol. 34, no. 1, pp. 17–29, 2000.
- [11] W. T. Chen, K. Huang, and M. N. Ardiansyah, "A mathematical programming model for aircraft leasing decisions," *Journal of Air Transport Management*, vol. 69, pp. 15–25, 2018.
- [12] P. Kuhle, D. Arroyo, and E. Schuster, "Building A blockchain-based decentralized digital asset management system for commercial aircraft leasing," *Computers in Industry*, vol. 126, Article ID 103393, 2021.
- [13] BTS, "Share of Leased Aircraft in the Aviation Industry Worldwide from 1970 to 2020," 2021, <https://www.statista.com/statistics/1095749/share-leased-aircraft-aviation-industry-worldwide/>.
- [14] Statista, "Size of the global aircraft leasing market from 2021 to 2029," 2022, <https://www.statista.com/statistics/747420/worldwide-commercial-aircraft-leasing-market-size/>.
- [15] E. Endrizalova, M. Novak, V. Nemeč, J. Hyrslova, and P. Mrazek, "Operating lease as a specific form of airlines outsourcing," 2018, <https://hrcak.srce.hr/ojs/index.php/plum/article/view/7916>.
- [16] P. Belobaba, "The airline planning process," *The Global Airline Industry*, pp. 153–181, 2009.
- [17] Q. Zhang, F. T. Chan, S. H. Chung, and X. Fu, "Operational aircraft maintenance routing problem incorporating cruise speed control," *Engineering Optimization*, pp. 1–20, 2022.
- [18] X. F. Liu, Z. H. Zhan, J. D. Deng, Y. Li, T. Gu, and J. Zhang, "An energy efficient ant colony system for virtual machine placement in cloud computing," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 1, pp. 113–128, 2018.
- [19] Z. G. Chen, Z. H. Zhan, Y. Lin et al., "Multiobjective cloud workflow scheduling: a multiple populations ant colony system approach," *IEEE Transactions on Cybernetics*, vol. 49, no. 8, pp. 2912–2926, 2019.
- [20] D. Liang, Z. H. Zhan, Y. Zhang, and J. Zhang, "An efficient ant colony system approach for new energy vehicle dispatch problem," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 11, pp. 4784–4797, 2020.
- [21] X. Zhang, Z. H. Zhan, W. Fang, P. Qian, and J. Zhang, "Multipopulation ant colony system with knowledge-based local searches for multiobjective supply chain configuration," *IEEE Transactions on Evolutionary Computation*, vol. 26, no. 3, pp. 512–526, 2022.
- [22] R. Chandren Muniyandi, M. K. Hasan, M. R. Hammoodi, and A. Maroosi, "An improved harmony search algorithm for proactive routing protocol in VANET," *Journal of Advanced Transportation*, vol. 2021, Article ID 6641857, 17 pages, 2021.
- [23] L. F. D. M. López, F. S. García, J. E. Naranjo Hernández, and N. G. Blas, "Speed proportional integrative derivative controller: optimization functions in metaheuristic algorithms," *Journal of Advanced Transportation*, vol. 2021, Article ID 5538296, 12 pages, 2021.
- [24] R. Costa Affonso, F. Couffin, and P. Leclaire, "Modelling of user behaviour for static rebalancing of bike sharing system: transfer of demand from bike-shortage stations to neighbouring stations," *Journal of Advanced Transportation*, vol. 2021, Article ID 8825521, 15 pages, 2021.
- [25] L. Shi, Z. H. Zhan, D. Liang, and J. Zhang, "Memory-based ant colony system approach for multi-source data associated dynamic electric vehicle dispatch optimization," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 10, pp. 17491–17505, 2022.
- [26] J. Y. Li, X. Y. Deng, Z. H. Zhan et al., "A multipopulation multiobjective ant colony system considering travel and prevention costs for vehicle routing in COVID-19-like epidemics," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 12, pp. 25062–25076, 2022.
- [27] L. J. Wu, Z. G. Chen, C. H. Chen et al., "Real environment-aware multisource data-associated cold chain logistics scheduling: a multiple population-based multiobjective ant colony system approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 12, pp. 23613–23627, 2022.
- [28] R. Wang, F. Ji, Y. Jiang et al., "An adaptive ant colony system based on variable range receding horizon control for berth allocation problem," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 11, pp. 21675–21686, 2022.
- [29] L. J. Wu, L. Shi, Z. H. Zhan, K. K. Lai, and J. Zhang, "A buffer-based ant colony system approach for dynamic cold chain logistics scheduling," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 6, no. 6, pp. 1438–1452, 2022.
- [30] H. Zhang and S. Ni, "Train scheduling optimization for an urban rail transit line: a simulated-annealing algorithm using a large neighborhood search metaheuristic," *Journal of Advanced Transportation*, vol. 2022, Article ID 9604362, 17 pages, 2022.
- [31] J. Zhuang, S. C. Chu, C. C. Hu, L. Liao, and J. S. Pan, "Advanced phasmatodea population evolution algorithm for capacitated vehicle routing problem," *Journal of Advanced Transportation*, vol. 2022, Article ID 9241112, 20 pages, 2022.

- [32] A. E. Eltoukhy, Z. Wang, F. T. Chan, S. H. Chung, H. L. Ma, and X. Wang, "Robust aircraft maintenance routing problem using a turn-around time reduction approach," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 50, no. 12, pp. 4919–4932, 2020.
- [33] A. E. Eltoukhy, Z. X. Wang, F. T. Chan, and X. Fu, "Data analytics in managing aircraft routing and maintenance staffing with price competition by a Stackelberg-Nash game model," *Transportation Research Part E: Logistics and Transportation Review*, vol. 122, pp. 143–168, 2019.
- [34] H. M. Naimi and N. Taherinejad, "New robust and efficient ant colony algorithms: using new interpretation of local updating process," *Expert Systems with Applications*, vol. 36, no. 1, pp. 481–488, 2009.
- [35] A. Sarac, R. Batta, and C. M. Rump, "A branch-and-price approach for operational aircraft maintenance routing," *European Journal of Operational Research*, vol. 175, no. 3, pp. 1850–1869, 2006.
- [36] R. Gopalan and K. T. Talluri, "The aircraft maintenance routing problem," *Operations Research*, vol. 46, no. 2, pp. 260–271, 1998.
- [37] Z. Liang and W. A. Chaovalitwongse, "The aircraft maintenance routing problem," *Optimization and Logistics Challenges in the enterprise*, vol. 04, pp. 327–348, 2009.
- [38] L. Marla, B. Vaaben, and C. Barnhart, "Integrated disruption management and flight planning to trade off delays and fuel burn," *Transportation Science*, vol. 51, no. 1, pp. 88–111, 2017.
- [39] H. Gürkan, S. Gürel, and M. S. Aktürk, "An integrated approach for airline scheduling, aircraft fleet and routing with cruise speed control," *Transportation Research Part C: Emerging Technologies*, vol. 68, pp. 38–57, 2016.
- [40] EUROCONTROL, "User Manual for the Base of Aircraft Data (BADA) Revision 3.11," Technical Report 13/04/16-01, Bretigny-sur-Orge, France, 2013.
- [41] EUROCONTROL, "Forecasting civil aviation fuel burn and emissions in Europe," Technical Report 2001-8, Bretigny-sur-Orge, France, 2001.
- [42] BTS, "Airline On-Time Performance Data," 2020, <https://www.transtats.bts.gov/ONTIME/>.
- [43] A. S. Duran, S. Gürel, and M. S. Aktürk, "Robust airline scheduling with controllable cruise times and chance constraints," *IIE Transactions*, vol. 47, no. 1, pp. 64–83, 2015.
- [44] M. S. Aktürk, A. Atamtürk, and S. Gürel, "Aircraft rescheduling with cruise speed control," *Operations Research*, vol. 62, no. 4, pp. 829–845, 2014.