Robust weekly aircraft maintenance routing problem and the extension to the tail assignment problem

Zhe Liang\textsuperscript{a}, Yuan Feng\textsuperscript{b}, Xiaoning Zhang\textsuperscript{a,\*}, Tao Wu\textsuperscript{c}, Wanpracha Art Chaovalitwongse\textsuperscript{d}

\textsuperscript{a} School of Economics \& Management, Tongji University, Shanghai 200092, China
\textsuperscript{b} Department of Industrial Engineering \& Management, Peking University, Beijing 100871, China
\textsuperscript{c} Advanced Analytics, Dow Chemical, Midland, MI 48674, USA
\textsuperscript{d} Departments of Industrial \& Systems Engineering and Radiology, University of Washington, Seattle, WA 98195, USA

Abstract

In this paper, we study two closely related airline planning problems: the robust weekly aircraft maintenance routing problem (RWAMRP) and the tail assignment problem (TAP). In real life operations, the RWAMRP solution is used in tactical planning whereas the TAP solution is implemented in operational planning. The main objective of these two problems is to minimize the total expected propagated delay (EPD) of the aircraft routes. To formulate the RWAMRP, we propose a novel weekly line-of-flights (LOF) network model that can handle complex and nonlinear cost functions of EPD. Because the number of LOFs grows exponentially with the number of flights to be scheduled, we propose a two-stage column generation approach to efficiently solve large-scale real-life RWAMRPs. Because the EPD of an LOF is highly nonlinear and can be very time-consuming to accurately compute, we propose three lower bounds on the EPD to solve the pricing subproblem of the column generation. Our approach is tested on eight real-life test instances. The computational results show that the proposed approach provides very tight LP relaxation (within 0.6\% of optimal solutions) and solves the test case with more than 6000 flights per week in less than three hours. We also investigate the solutions obtained by our approach over 500 simulated realizations. The simulation results demonstrate that, in all eight test instances, our solutions result in less EPDs than those obtained from traditional methods. We then extend our model and solution approach to solve realistically simulated TAP instances.

1. Introduction

As commercial aviation celebrates its first century in 2014, the development of aviation has brought enormous economic growth and spread prosperity globally through air connectivity. The demand of commercial aviation is currently blooming with the recovery of the global economics. In 2013, the total international and domestic passenger demand grew 5.2\% to a new high of 3.1 billion passengers (ICAO, 2014). IATA Airline Industry Forecast 2013–2017 shows that the passenger demand is expected to be 3.91 billion passengers in year 2017, a 31\% increase from year 2012 (IATA, 2014). Despite the tremendous benefits from the commercial aviation, the rapid growth of air traffic has brought many difficult challenges in planning and
operations of the airline industry. For example, in year 2011, 103 million system delay minutes were estimated to have $7.7 billion USD in direct aircraft operating costs for scheduled U.S. passenger airlines (A4A, 2011). With the predicted growth of future air traffic, such irregular flight operations are likely to have a deeper impact on the airline industry.

Over the last two decades, researchers have increasingly focused on how to manage irregular airline operations, in responding to the rising cost of flight cancelation and delay by disruptions. There are two main schemes to address this issue: reactive scheme and proactive scheme. After the planned schedule of the aircraft, crews, and passengers is disrupted by irregular flight operations, the reactive scheme (also known as disruption management) attempts to recover the schedule at a minimum disruption cost. On the other hand, the proactive scheme (also known as the robust scheduling) attempts to construct a robust schedule of the aircraft, crews, and passengers that is resilient to flight disruptions. There are two main approaches for robust scheduling. The first approach is to absorb disruptions in the schedule by placing an adequate time buffer between connecting flights so that short disruptions will not propagate through out the schedule. The second approach is to create a schedule with an easy recovery by constructing aircraft rotation and crew pairing solutions that are flexible and provide many alternatives for swapping and cancelation of the aircraft and its crews.

In this paper, we focus on the disruption absorbing approach, which is to create a robust routing/schedule of the aircraft while taking into account the expected propagated delay (EPD) of each flight sequence by allocating the optimal flight buffer to minimize the EPD based on the historical distribution of non-propagated delays. This optimization problem is called a robust weekly aircraft maintenance routing problem (RWAMRP). To formulate the RWAMRP, we develop a novel weekly line-of-flights (LOF) network model, which is derived from an integration of the state-of-the-art flight string model (Barnhart et al., 1998; Lan et al., 2006) with the compact network model of aircraft rotation tour (Liang et al., 2011; Liang and Chaovalitwongse, 2013). Because the number of LOFs grows exponentially with the number of flights to be scheduled, we propose a two-stage column generation approach to efficiently solve large-scale, practical RWAMRPs. In addition, we also extend our approach to model and solve the operational tail assignment problem (TAP). Computational results suggest that our approach can provide more robust aircraft routing solutions with less propagated delay than existing approaches. The main contributions of our paper can be listed as follows.

- The new weekly line-of-flights (LOF) network that we propose can model complex and nonlinear cost functions, including the EPD. Our model is very compact and contains a much smaller number of the variables than that of the flight string model. Because our model is network-based, it provides a much tighter LP relaxation as also evidenced by the empirical result in this paper.
- The calculation of EPD that we propose offers a more accurate computation, which proves that the traditional EPD method in the literature (Lan et al., 2006; Dunbar et al., 2012) underestimates the EPD of the LOFs with more than two flights. The underestimation could be as much as 40% for some LOFs.
- Although our LOF model is more compact than the flight string model, it is impractical to enumerate all possible LOFs. In addition, accurate computation of EPD for an LOF can be very time consuming if the number of flights in the LOF is large. By proposing three lower bounds on the EPD of an LOF, our two-stage column generation efficiently generates only high-quality LOFs and computes the accurate EPDs only when it is truly necessary.
- The proposed LOF model can be generalized to solve the operational TAP, and it can also be modified to handle several other operational considerations.

The remainder of the paper is organized as follows: In Section 2, we give a brief review on the robust airline planning problems and on the aircraft maintenance routing problem. In Section 3, we propose the mathematical models for the RWAMRP and TAP respectively. In Section 4, we present a heuristics and a column generation framework to solve the proposed models optimally. In Section 5, we provide the computational experience for eight test cases. We conclude in the final section of this paper.

2. Background

2.1. Overview of airline planning and scheduling

Airline planning and scheduling generally consists of four major sequential operations: flight schedule design, fleet assignment, aircraft maintenance routing, and crew scheduling. The flight schedule is usually designed by the airline marketing department based on traffic forecasts, airline network analysis, and profitability analysis over several months (Soumis et al., 1980; Phillips et al., 1991; Pilla et al., 2014). Subsequently, based on the designed flight schedule, a variety of aircraft fleets are assigned to individual flights according to passenger demands, revenues, operating costs, etc., such that the total profit is maximized (Hane et al., 1995; Sherali et al., 2006; Belanger et al., 2006). Given an assigned aircraft fleet, (Clarke et al., 1997; Barnhart et al., 1998; Liang et al., 2011; Liang and Chaovalitwongse, 2013) a rotation of flight schedules for each individual aircraft is constructed while ensuring that there are enough maintenance opportunities. Subsequently, the crew scheduling is to determine the best set of crew pairings (i.e., crew trips spanning one or more working days separated by a break period) to cover all the aircraft fleets (Ryan and Foster, 1981; Desaulniers et al., 1997a; Vance et al., 1997; Boubaker et al., 2010) and to construct personalized monthly schedules (rosters) for crew members. Traditionally, these problems are solved sequentially, and the result of an upstream problem is fed into the next downstream problem. This sequential method...
may lead to sub-optimal solutions to the overall operations. Therefore, in the last decade, researchers have started to focus on integrated problems where more than one planning operation are solved simultaneously (Barnhart et al., 1998; Stojkovic and Soumis, 2001; Cohn and Barnhart, 2003; Mercier et al., 2005; Papadakos, 2009; Sherali et al., 2013).

2.2. Robust airline planning

As the schedules generated by airline planning are vulnerable to operational disruptions, researchers have recently begun studies in robust airline planning. For example, the airline planning should construct a robust flight schedule that can either absorb disruptions or provide more convenient recovery opportunities such as swap aircraft and crews, and short cycle cancelation of flights, etc. An overview of research investigations in robust airline planning for each of the major airline planning problems is illustrated in Table 1.

Unlike the traditional schedule design, most robust schedule design problems are solved after the AMR and/or CS so that the feasibility for aircraft and/or crews are preserved, and the planned costs for aircraft rotation and crew pairing are unaffected. Schaefer and Nemhauser (2006) first proposed a mathematical model to retime the schedule while preserving the feasibility of crew pairings. A multi-objective genetic algorithm (Lee et al., 2007) and subsequently a mathematical model (Ahmadbeygi et al., 2010) were developed to improve the robustness of the schedule by retiming flights departure time while maintaining the original connections for the aircraft and crews. Burke et al. (2010) proposed a memetic algorithm to maximize the sum of the reliability for all aircraft connections while preserving the feasibility of aircraft rotation without considering crew connections. Most recently, Sohoni et al. (2011) proposed two stochastic programming models to solve the schedule design problem with the consideration of block-time uncertainty.

As the fleet assignment and aircraft rotation with many short cycles are less sensitive to flight cancelations, Rosenberger et al. (2004) proposed an integer programming model to solve the FAM with less hub connectivity (hub isolation) and more short cycles. A cycle is a sequence of flights in an aircraft rotation in which the first flight departs from the same airport at

<table>
<thead>
<tr>
<th>Paper</th>
<th>SD</th>
<th>AR</th>
<th>CS/CP</th>
<th>PI</th>
<th>Methodology</th>
<th>Mechanism</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rosenberger et al. (2004)</td>
<td>✓</td>
<td></td>
<td></td>
<td>IP</td>
<td>Hub isolation &amp; short cycles</td>
<td></td>
</tr>
<tr>
<td>Shebalov and Klabjan (2006)</td>
<td></td>
<td>✓</td>
<td></td>
<td>IP &amp; CG</td>
<td>Move-up crews</td>
<td></td>
</tr>
<tr>
<td>Yen and Birge (2006)</td>
<td></td>
<td></td>
<td></td>
<td>SP</td>
<td>Retiming flights to minimize PD</td>
<td></td>
</tr>
<tr>
<td>Lan et al. (2006)</td>
<td></td>
<td></td>
<td></td>
<td>IP &amp; CG</td>
<td>Station purity</td>
<td></td>
</tr>
<tr>
<td>Smith and Johnson (2006)</td>
<td></td>
<td>✓</td>
<td></td>
<td>IP &amp; CG</td>
<td>Increase block time while maintaining feasibility for aircraft, crews, and passengers</td>
<td></td>
</tr>
<tr>
<td>Schaefer and Nemhauser (2006)</td>
<td></td>
<td>✓</td>
<td></td>
<td>IP</td>
<td>Retiming to minimize the delay propagation</td>
<td></td>
</tr>
<tr>
<td>Lee et al. (2007)</td>
<td></td>
<td>✓</td>
<td></td>
<td>IP</td>
<td>Retiming flights to minimize PD &amp; Short cycles</td>
<td></td>
</tr>
<tr>
<td>Gao et al. (2009)</td>
<td></td>
<td></td>
<td></td>
<td>IP</td>
<td>Retiming to minimize the delay propagation</td>
<td></td>
</tr>
<tr>
<td>Tekiner et al. (2009)</td>
<td></td>
<td></td>
<td></td>
<td>IP</td>
<td>Crew pairing to manage extra flights besides regular flight schedule</td>
<td></td>
</tr>
<tr>
<td>Ahmadbeygi et al. (2010)</td>
<td></td>
<td></td>
<td></td>
<td>IP</td>
<td>Crew pairing to manage extra flights besides regular flight schedule</td>
<td></td>
</tr>
<tr>
<td>Burke et al. (2010)</td>
<td></td>
<td>✓</td>
<td></td>
<td>IP</td>
<td>Retiming to maximize the reliability of the flight connection and the possible aircraft swaps</td>
<td></td>
</tr>
<tr>
<td>Weide et al. (2010)</td>
<td></td>
<td>✓</td>
<td></td>
<td>IP</td>
<td>Ensuring crews stay at the same aircraft at short connects, and minimizing the penalty for restricted connects</td>
<td></td>
</tr>
<tr>
<td>Dunbar et al. (2012)</td>
<td></td>
<td>✓</td>
<td></td>
<td>IP &amp; CG</td>
<td>Minimizing the total PD considering both aircraft delays and crew delays</td>
<td></td>
</tr>
<tr>
<td>Sohoni et al. (2011)</td>
<td></td>
<td></td>
<td></td>
<td>IP &amp; Cut</td>
<td>Considering block time uncertainty</td>
<td></td>
</tr>
<tr>
<td>Lapp and Cohn (2012)</td>
<td></td>
<td>✓</td>
<td></td>
<td>IP &amp; CG</td>
<td>Improve maintenance reachability at maintenance disruptions</td>
<td></td>
</tr>
<tr>
<td>Muter et al. (2013)</td>
<td></td>
<td>✓</td>
<td></td>
<td>IP &amp; CG &amp; Cut</td>
<td>Crew pairing to manage extra flights besides regular flight schedule</td>
<td></td>
</tr>
<tr>
<td>Jiang and Barnhart (2013)</td>
<td></td>
<td>✓</td>
<td></td>
<td>IP &amp; CG</td>
<td>Retiming of flight schedules with the de-banking consideration</td>
<td></td>
</tr>
<tr>
<td>Sherali et al. (2013)</td>
<td></td>
<td>✓</td>
<td></td>
<td>IP &amp; BD</td>
<td>Retiming and schedule balance</td>
<td></td>
</tr>
<tr>
<td>This paper</td>
<td></td>
<td>✓</td>
<td></td>
<td>IP &amp; CG</td>
<td>Minimizing the total Expected PD</td>
<td></td>
</tr>
</tbody>
</table>

SD: Schedule Design
FAM: Fleet Assignment Problem
AR: Aircraft Rotation Problem
CS/CP: Crew Scheduling/Pairing
PI: Passenger Itinerary
GA: Genetic Algorithm
MA: Memetic Algorithm
IP: Integer Programming
CG: Column Generation
SP: Stochastic Programming
Cut: Cutting Plan
BD: Benders Decomposition
which the last flight lands. Smith and Johnson (2006) solved the FAM while ensuring the station purity, that is, to limit the number of fleet types allowed at each airport in the schedule. Jiang and Barnhart (2013) proposed a de-banked retiming FAM that maximizes potential connecting itineraries. Sherali et al. (2013) presented a robust FAM model considering flight retiming, and passenger demand recapture, and solved it with a Benders decomposition approach.

Lan et al. (2006) is among the first studies that consider propagated delay (PD) for aircraft. They modified the string model for AMR to minimize the total PD. A column generation algorithm is used to solve the model, and a heuristic is proposed for the pricing subproblem. The authors also proposed a retiming model to minimize the total passenger misconnections. Lapp and Cohn (2012) presented an integer programming model to build a set of lines-of-flights (LOFs), a sequence of flights flown by an aircraft daily, such that the aircraft maintenance reachability is maximized.

Yen and Birge (2006) proposed a stochastic programming model for the CS problem to minimize the PD. The objective of the model takes into account random delays that are associated with the interactions between crews, aircraft, and pairings. Shebalov and Klabjan (2006) proposed a crew pairing model that maximizes not only the crew cost, but also the number of move-up crews, i.e., the crews that can potentially be swapped in operations when disruptions occur. Tekiner et al. (2009) proposed a robust crew pairing model such that the crew pairing solution can manage a set of extra flights besides the regular flight schedule. Muter et al. (2013) studied the same problem as in Tekiner et al. (2009), and proposed a column generation algorithm to solve the problem.

In the last five years, there has been an increasing interest in solving the integrated problems of two or more planning operations with the robustness consideration. Gao et al. (2009) proposed a robust integrated FAM and CS model that incorporates station purity and crew-base purity. Weide et al. (2010) presented a robust integrated model to solve aircraft routing problem and CP simultaneously. The integrated model ensures that the crew has to stay on the same aircraft at a short connection, and penalizes the restricted connections in the objective function. The short connection is a crew connection less than the minimum sit time for crews but longer than the minimum turn time for aircraft. Dunbar et al. (2012) also presented a robust integrated model to solve the aircraft routing problem and CP at the same time. The model minimizes the total PD induced by both aircraft and crews, i.e., the longer delay caused by either aircraft or crews because the flight delay is transferred between late running of aircraft and crew.

2.3. Aircraft maintenance routing problem (AMRP)

Given a set of flights and a fleet of aircraft, the AMRP is to determine the flight routes for every aircraft such that the maintenance requirements, which are set by Federal Aviation Administration (FAA) and individual airline companies, are satisfied. The aircraft maintenance considered in AMRP are called daily checks, which are needed every two to four days. It includes a walk around inspection, a checkup on lights, emergency equipment, servicing engine oil, etc., and repairs if needed. It normally lasts from one to three hours for a checkup, and several hours for a repair. In order not to affect the aircraft utilization rate during the day, a daily check is usually performed at night (Talluri, 1998; Gopalan and Talluri, 1998; Sriram and Haghani, 2003). Only certain airports, called maintenance stations, are capable of performing the maintenance operation. A feasible solution to the AMRP normally contains a generic aircraft route during a rolling time horizon. This generic solution does not assign individual aircraft to flights explicitly. However, in the operational stage, this solution can serve as a reference for assigning individual aircraft.

The solution methodology for the AMRP can be categorized into three approaches. The most commonly used approach in the literature, which was first proposed in Desaulniers et al. (1997b) and Barnhart et al. (1998), is to model a sequence of aircraft rotation as connecting flight strings and find the optimal routing by solving a set partitioning problem. This model has been further modified and extended in Cordeau et al. (2001), Elf and Kaibel (2003), Mercier et al. (2005), and Mercier and Soumis (2007). In this approach, the decision variables represent the maintenance feasible flight sequences between two maintenance stations with a maintenance at the end. Various column generation and branch-and-price solution approaches are developed to solve this type of models. The second approach models the AMRP as an Euler tour problem or asymmetric traveling salesman problem with side constraints (Clarke et al., 1997; Gopalan and Talluri, 1998; Talluri, 1998; Boland et al., 2000). The last and most recent approach models the AMRP as a network flow problem (Liang et al., 2011; Liang and Chaovalitwongse, 2013). The network model has been shown to be very compact and scalable as it is able to solve large real life daily and weekly AMRPs in a reasonable time. However, it is important to note that the network models in Liang et al. (2011) and Liang and Chaovalitwongse (2013) can only capture a linear cost function, but not the complex cost function such as PD as in the flight string model (Lan et al., 2006). On the other hand, the string based model contains much more variables than the network models in Liang et al. (2011) and Liang and Chaovalitwongse (2013), and column generation has to be used to solve the model. However, it is very difficult to prove the optimality of the column generation solutions because the expected propagated delay (EPD) of a string is highly non-linear and it is hard to solve the subproblem optimally in the column generation (Lan et al., 2006).

2.4. Extension to tail assignment problem (TAP):

For major airlines, the AMRP is often considered as a tactical problem to investigate the feasibility for an aircraft fleet to cover planned flights and satisfy the FAA maintenance requirements. Because there is no exact tail assignments decided at the operational level in the AMRP solution, it is not straightforward and can be very difficult to manage individual aircrafts in
a fleet at an operational level. The TAP is often solved to manage and assign the route for each individual aircraft in the fleet, and usually solved much closer to the day of operations (Sarac et al., 2006; Lapp and Wikenhauser, 2012; Jacobs et al., 2012; Basdere and Bilge, 2013). On average only 80–85% of the daily LOFs from the AMRP solution can be carried out in real operations. Because of the close relationship between the AMRP and the TAP, solution approaches for the AMRP can be seamlessly extended to solve the TAP by adding operational considerations. For example, Sarac et al. (2006) proposed a flight string model for the TAP that maximizes the remaining legitimate flying time for a set of critical aircrafts, in which an aircraft is considered to be critical if its accumulated flying hours reach a predefined level. Lapp and Wikenhauser (2012) proposed a TAP model that minimizes the total aircraft fuel consumption and emission. Basdere and Bilge (2013) proposed a connection-network model for the TAP that maximizes the utilization of the total remaining flying time of aircraft fleet.

3. The robust weekly AMR problem (RWAMRP)

The RWAMRP studied in this paper can be defined as follows. Given a weekly schedule containing $F$ flights, the objective of the RWAMRP is to find a set of aircraft routes that minimizes the overall propagated delay cost subject to (1) the number of aircrafts used in the set of aircraft routes has to be less than or equal to the fleet size $K$; (2) each aircraft needs to undergo a maintenance every $D$ days or less; (3) maintenances can only be performed at one of the maintenance stations denoted by $M$; (4) maintenance stations are capacitated, with capacity $Q_mj$ on $j$ day of the week at maintenance station $m$, where $j \in J = \{1, 2, \ldots, 7\}$ and $m \in M$; and (5) the aircraft rotation should be cyclic, that is the number of aircraft in each station at the beginning and end of the week are the same, so that the schedule can be repeated week after week.

To formulate the RWAMRP, we herein propose a new mathematical programming model that hybridizes the flight string model in Desaulniers et al. (1997b) and Barnhart et al. (1998) and the aircraft rotation network model in Liang et al. (2011) and Liang and Chaovalitwongse (2013). The premise of our model is to generate LOFs that last a single day instead of generating flight strings that last for several days because the delay propagation normally ends at night. The number of possible LOFs, which usually contain at most five or six flights, is much less than the number of flight strings that could last for three or four days and contain up to 20 flights. Based on the set of LOFs generated, we construct a weekly LOF network model, called robust weekly LOF network model (RWLNM). We also provide an alternative cost structure for the RWLNM, and extend it to solve the operational TAP. Finally, it is worth mentioning that for some international flights the delay might propagate overnight. Therefore, the models we developed are more suitable for domestic airlines.

3.1. Weekly LOF network (WLN)

The first step to formulate the RWLNM is to construct a WLN from flight information. In this study, we extend the weekly rotation tour network, as proposed in Liang and Chaovalitwongse (2013), to construct the WLN. As shown in Fig. 2, the WLN contains two types of nodes: start-day nodes and end-day nodes. A start-day node represents the starting of the day at a station, and an end-day node represents the end of the day at a station. The WLN contains three types of arcs: LOF arcs, night arcs, and maintenance arcs. An LOF arc connects a start-day node and an end-day node of the same day, representing a possible LOF for an aircraft. A possible LOF contains a sequence of flights that can be flown by an aircraft within a day. For example, in Fig. 1a, a one-day schedule contains 8 flights between 3 stations, and a number of possible LOFs are presented in Fig. 1b. A night arc connects an end-day node of the previous day and the start-day node of the next day at the same station, representing aircraft overnight at the station. With LOFs and night arcs, we construct a $D$-day subnetwork, which starts at day $j$ and ends at day $j + D - 1$, where $j \in \{1, 2, \ldots, 7\}$. If $j + D - 1$ is greater than 7, we divide it by 7 and take the remainder to ensure that the day index is smaller than or equal to 7. Without loss of generality, we assume all the day indices calculated in the remaining of the paper are smaller than or equal to 7 by using the modular operation as mentioned previously. We denote the $D$-day subnetwork starting at day $j$ of the week, where $j \in \{1, 2, \ldots, 7\}$, as $S_j$. For example, subnetwork $S_5$, where $D = 3$, is the subnetwork containing Saturday, Sunday, and Monday. It is noted that we need to construct duplicated $D$ LOF arcs for a single LOF departing on day $i + D - 1$, each in one of $S_j$ subnetwork where $j \in \{i, \ldots, i + D - 1\}$. For example, we

![Fig. 1. Construction of LOFs. In (a) a one-day schedule contains 8 flights between 3 stations. In (b) a number of possible LOFs are presented.](image-url)
need to construct 3 LOF arcs for an LOF on Saturday in subnetworks S_4, S_5, and S_6 respectively, because all these subnetworks contain the LOFs on Saturday. We also create a set of capacitated maintenance arcs connecting seven S_j subnetworks. Specifically, a maintenance arc starts at the end-day node i at maintenance station m in S_j subnetwork, where $i \in \{j \ldots j + D - 1\}$, and ends at the beginning start-day node at maintenance station m in subnetwork S_{j+1}. For example, as shown in Fig. 2, for subnetwork S_2 (here $D = 3$) spanning on Tuesday, Wednesday, and Thursday, we create three maintenance arcs leaving this subnetwork at the end of Tuesday, Wednesday, and Thursday. The maintenance arc, which starts at the end of Tuesday, terminates at the beginning of subnetwork S_3; and the maintenance arc, which starts at the end of Wednesday, terminates at the beginning of subnetwork S_4.

By constructing the WLN, it is guaranteed that no aircraft rotations violate the maximum $D$-day maintenance constraints. This is because without going through any maintenance arcs, an aircraft only stays in a single $S_p$ subnetwork in WLN, which lasts for $D$ days. In order to build a flight route lasting more than $D$ days, a maintenance arc has to be inserted in the route. In other words, any flow in the WLN must traverse a maintenance arc after $d \in \{1 \ldots D\}$ days to respect the maintenance requirements. For any maintenance station of a $S_j$ subnetwork, there are $D$ outgoing maintenance arcs, each at the end of day $i \in \{j \ldots j + D - 1\}$. Therefore, an aircraft can receive a maintenance after $d \in \{1 \ldots D\}$ days of flying, hence no maintenance opportunities are ignored in the network.

### 3.2. Expected propagated delay cost of LOF

The main objective for our RWAMRP is to minimize the total EPD of the aircraft rotations. The flight delays can be divided into two categories: non-propagated delay and propagated delay. A non-propagated delay is caused by issues that are not a function of routing, such as mechanical problems, bad weather, safety issues, passenger delays, etc. A propagated delay...
occurs when an aircraft of a later flight is delayed because its previous flight is delayed. Since non-propagated delays are not affected by the aircraft routing decisions, we only consider the propagated delay in RWAMRP. Because it is logical to assume that the propagated delays end at night, the total EPD of an aircraft rotation can be decomposed into a set of LOFs. However, the computation of the EPD for LOFs is nontrivial. To simplify the computation of EPD, we assume that non-propagated delays of different flights in an LOF are independent, because it is out of the scope of this paper to study the correlations of non-propagated delays. In this section, we discuss how to accurately estimate the expected propagated delay cost for an LOF. To facilitate our discussion, we define the following notations:

\[ t_f: \text{Non-propagated delay (NPD) (in minute) of an aircraft for flight } f. \text{ We assume } t_f \in \{0, 1, \ldots, T\}, \text{ where } T \text{ is the upper bound for NPD.} \]

\[ p_{t_f}: \text{The probability for an aircraft to have } t \text{ minutes of NPD for flight } f. \]

\[ b_{t_f}: \text{The buffer time between two connecting flights } f \text{ and } f'. \text{ in an LOF.} \]

The EPD of \( f \), and the EPD of an LOF with \( n \) flights \( f_1, \ldots, f_n \) can be recursively computed as follows (Lan, 2003; Lan et al., 2006; Dunbar et al., 2012):

\[
\begin{align*}
\text{EPD}_{f_1} &= \sum_{t_{f_1}=0}^{T} (t_{f_1} - b_{f_1})^+ p_{t_{f_1}} \quad (1) \\
\text{EPD}_{f_2} &= \sum_{t_{f_2}=0}^{T} \left( \text{EPD}_{f_1} + t_{f_2} - b_{f_1} \right)^+ p_{t_{f_2}} \quad (2) \\
& \vdots \\
\text{EPD}_{f_n} &= \sum_{t_{f_n}=0}^{T} \left( \text{EPD}_{f_{n-1}} + t_{f_n} - b_{f_{n-1}} \right)^+ p_{t_{f_n}} \quad (3) \\
\text{c}_l &= \sum_{t_{i}=2}^{n} \text{EPD}_{f_i} \quad \forall l \in L. \quad (4)
\end{align*}
\]

Here, \((t_f - b_{t_f})^+ = \max\{t_f - b_{t_f}, 0\}\). The advantage of the method is that it can compute the total EPD of an LOF efficiently because the EPD of the next flight only depends on the EPD of the previous flight as shown in Eqs. (1)–(4). However, this method underestimates the real value of the EPD of an LOF. In fact, the EPD can be more accurately computed as follows:

\[
\begin{align*}
\text{EPD}_{f_1} &= \sum_{t_{f_1}=0}^{T} (t_{f_1} - b_{f_1})^+ p_{t_{f_1}} \quad (5) \\
\text{EPD}_{f_2} &= \sum_{t_{f_2}=0}^{T} \sum_{t_{f_1}=0}^{T} \left( (t_{f_1} - b_{f_1})^+ + t_{f_2} - b_{f_1} \right)^+ p_{t_{f_1}} p_{t_{f_2}} \quad (6) \\
& \vdots \\
\text{EPD}_{f_n} &= \sum_{t_{f_n}=0}^{T} \cdots \sum_{t_{f_{n-1}}=0}^{T} \left( \cdots (t_{f_1} - b_{f_1})^+ + \cdots + t_{f_{n-1}} - b_{f_{n-1}} \right)^+ p_{t_{f_1}} \cdots p_{t_{f_{n-1}}} \quad (7) \\
\text{c}_l &= \sum_{t_{i}=2}^{n} \text{EPD}_{f_i} \quad \forall l \in L. \quad (8)
\end{align*}
\]

Here we provide a simple numerical example to show the difference between the above two methods. A detailed proof is given in Appendix A. Consider an LOF with three flights \( A \rightarrow B \rightarrow C \). The buffer periods between flights \( A, B \) and flights \( B, C \) are both 15 min. Assume that the probability function of NPDs for all flights are the same, as shown in Table 2. Therefore, \( \text{EPD}_B \) for flight B can be computed as \( \text{EPD}_B = \sum t_{a} \in \{0, \ldots, 50\} (t_a - 15)^+ p_{t_a} = 7.5 \) min. By using Eq. (2), the EPD for flight \( C \) is computed as \( \text{EPD}_C = \sum t_{a} \in \{0, \ldots, 50\} (7.5 + t_a - 15)^+ p_{t_a} = 12. \) However, by using Eq. (6), the EPD for flight \( C \) is \( \sum t_{a} \in \{0, \ldots, 50\} \sum t_{b} \in \{0, \ldots, 50\} (t_a - 15)^+ + t_b - 15)^+ p_{t_a} p_{t_b} = 13.15. \) From this example, we clearly see that the total EPD is underestimated using Eqs. (1)–(4). In our computational study, we find that the underestimation of EPD for some LOFs could be as

<table>
<thead>
<tr>
<th>NPD d (in mins)</th>
<th>0</th>
<th>10</th>
<th>20</th>
<th>30</th>
<th>40</th>
<th>50</th>
</tr>
</thead>
<tbody>
<tr>
<td>Probability ( p_d )</td>
<td>20%</td>
<td>30%</td>
<td>20%</td>
<td>15%</td>
<td>10%</td>
<td>5%</td>
</tr>
</tbody>
</table>
much as 40%, which is quite significant because the reported savings by robust planning are usually from 15% to 45% (Lan et al., 2006; Dunbar et al., 2012). In the optimal solution of 8 test cases, $c_i$ underestimates the totals EPD by average of 7%. Finally, it is worth to mention that this result can be easily generalized when flights have different non-propagated delay distribution as long as they are independent.

As we can see from Eqs. (5)–(8), the accurate computation of the EPD for LOFs is highly nonlinear. Specifically, the computational complexity for EPD of flight $j$ with $n$ proceeding flights is $O(T^n)$. Here, $n \leq 6$ since an aircraft normally fly 7 or fewer flights in a day from our analysis on the historical data. To speed up the computation of the EPD, we group the probability mass function into $T$ region where $T < T$, and we assume the uniform distribution within each region. By doing so, we reduce the complexity from $T^n$ to $T^n$. In Fig. 3a, we show an empirical non-propagated delay distribution for a fleet of a US domestic airline where $T = 240$. In Fig. 3b, we show the approximation distribution by approximating the distribution with 48 regions. Our computational results shown that the approximation using $T$ regions provide accurate EPD with less than 2% approximation error. Finally, it is worth mentioning that even with the reduction on $T$, it is still very time consuming to compute $c_i$ if the number of flights in an LOF is large. For example, it takes more than 10 min to compute $c_i$ for a single LOF with seven flights when $T = 48$.

3.3. Robust weekly LOF network model (RWLNM)

We define the following notations that will be used throughout the paper.

Sets, Elements, and Constants

- $D$: the maximum days between two consecutive maintenances.
- $j \in \{1, 2, \ldots, 7\}$ represents the day of week.
- $S_j$: the $D$-day subnetwork starting on day $j$.
- $N$: the set of nodes in the WLN, indexed by $n$.
- $N_j$: the set of node in $S_j$ subnetwork, where $j \in \{1, 2, \ldots, 7\}$.
- $F$: the set of flights in the weekly schedule, indexed by $f$.
- $L$: the set of LOFs in the weekly schedule, indexed by $l$.
- $c_l$: the cost of LOF $l \in L$. In the robust aircraft maintenance routing problem, $c_l$ is the sum of EPD for all the flights in LOF $l$.
- $L_j$: the set of LOF arcs in $S_j$ subnetwork. It is worth mentioning that $L_j$ are the arcs in a subnetwork, whereas $L$ are the LOFs in reality. Therefore, $\sum_{j \in (1, 7)} |L_j| = D \times |L|$, because each LOF appears $D$ times in WLN.
- $M$: the set of maintenance stations, indexed by $m$.
- $R$: the set of maintenance arcs in WLN, indexed by $r$.
- $r_{md}$: the maintenance arc at station $m$ at the end of day $j$ after $d$ days flying, where $m \in M, j \in (1, 7)$, and $d \in D$. Notice that index $(mjd)$ uniquely defines a maintenance arc in the WLN.
- $Q_{mj}$: the maintenance capacity at maintenance station $m$ on day $j$.
- $G$: the set of over night arcs in the WLN, indexed by $g$.
- $g_{md}$: the over night arc before node $n$.
- $g_{nd}$: the over night arc after node $n$.
- $K$: the size of the aircraft fleet.
- $O$: the arbitrary count time for counting the number of aircraft.
- $F_O$: the set of flight arcs passing the count time $O$ in WLN.
- $L_O$: the set of LOF arcs passing the count time $O$ in WLN.

![PDF of the historical non-propagated delay for a fleet of a US domestic airline](a) PDF of the historical non-propagated delay for a fleet of a US domestic airline

![Approximated PDF of the historical non-propagated delay](b) Approximated PDF of the historical non-propagated delay

Fig. 3. The PMF and approximation of the PMF for the non-propagated delay of a US domestic fleet.
$G_0$: the set of over night arcs passing the count time $O$ in WLN.
$R_0$: the set of maintenance arcs passing the count time $O$ in WLN.

### Indication Parameters

- $\lambda_{jl}$: the binary indicator such that $\lambda_{jl} = 1$ if LOF $l$ contains flight $f$, and 0 otherwise.
- $x_{jn}^+$: the binary indicator such that $x_{jn}^+ = 1$ if LOF $l$ in $S_j$ network starts at node $n$, and 0 otherwise.
- $x_{jn}^-$: the binary indicator such that $x_{jn}^- = 1$ if LOF $l$ in $S_j$ network ends at node $n$, and 0 otherwise.
- $\gamma_{gmjn}^+$: the binary indicator such that $\gamma_{gmjn}^+ = 1$ if over night arc $g$ in $S_j$ network starts at node $n$, and 0 otherwise.
- $\gamma_{gmjn}^-$: the binary indicator such that $\gamma_{gmjn}^- = 1$ if over night arc $g$ in $S_j$ network ends at node $n$, and 0 otherwise.
- $p_{mjdn}^+$: the binary indicator such that $p_{mjdn}^+ = 1$ if maintenance arc $g_{mjdn}$ starts at node $n$, and 0 otherwise.
- $p_{mjdn}^-$: the binary indicator such that $p_{mjdn}^- = 1$ if maintenance arc $g_{mjdn}$ ends at node $n$; and 0 otherwise.

### Variables

- $x_{lj}$: the binary variable such that $x_{lj} = 1$ if LOF $l$ is flown in $S_j$ subnetwork, and 0 otherwise.
- $y_k$: the integer variable representing the number of aircraft on the over night arc $g$ in the weekly network.
- $z_{mjdn}$: the integer variable representing the number of aircraft on maintenance arc $R_{mjdn}$ in the weekly network.

Based on the above notations, the robust WLN model (RWLNM) is given by

$$\min \sum_{j \in \{1, \ldots, 7\}} \sum_{k \in L_j} c_jx_{lj}$$

s.t.

$$\sum_{j \in \{1, \ldots, 7\}} \sum_{k \in L_j} \lambda_{jl}x_{lj} = 1 \quad \forall f \in F, \tag{9}$$

$$\sum_{j \in \{1, \ldots, 7\}} \sum_{k \in L_j} x_{jn}^+ + \sum_{m \in M_{D \cup D_j}} \sum_{j \in \{1, \ldots, 7\}} p_{mjdn}^+ z_{mjdn} + y_{jn} = \sum_{j \in \{1, \ldots, 7\}} \sum_{k \in L_j} x_{jn}^- + \sum_{m \in M_{D \cup D_j}} \sum_{j \in \{1, \ldots, 7\}} p_{mjdn}^- z_{mjdn} + y_{jn} \quad \forall n \in N, \tag{10}$$

$$\sum_{d \in D} z_{mjdn} \leq Q_{mj} \quad \forall m \in M, j \in \{1, \ldots, 7\}, \tag{11}$$

$$\sum_{l \in L_0} x_{lj} + \sum_{r_{mjdn} \cap k_n} z_{mjdn} + \sum_{g \in G_0} y_g \leq K \quad \tag{12}$$

$$x_{lj} \in \{0, 1\} \quad \forall j \in \{1, \ldots, 7\}, \forall l \in L_j, \tag{13}$$

$$z_{mjdn} \in \{0, 1, \ldots, Q_{mp}\} \quad \forall m \in M, \forall j \in \{1, \ldots, 7\}, \forall d \in D, \tag{14}$$

$$y_g \in Z^+ \quad \forall g \in G. \tag{15}$$

The objective function in Eq. (9) minimizes the total expected propagated delay cost. The assignment constraints in Eq. (10) ensure that each flight is covered once in the rotation solution. The flow balance constraints in Eq. (11) ensure that the number of inbound aircraft is equal to the number of outbound aircraft at each node. The capacity constraints in Eq. (12) ensure that the number of aircraft being maintained at each station is not greater than the station’s capacity. The fleet size constraint in Eq. (13) ensures that the total number of aircraft used is not greater than the size of the fleet. The constraints in Eqs. (14)–(16) are the binary, integrality constraints for variables. It is worth mentioning that the integrality constraints in Eqs. (15) and (16) can be relaxed because of flow balance constraints in Eq. (11) and binary constraints in Eq. (14).

### 3.4. tail assignment problem (TAP)

When it is close to the day of operations, due to possible disruptions the routes provided by the AMR are often modified and later assigned to individual aircrafts. This operational planning is called the operational tail assignment problem (TAP), which is to find the weekly routes for the set of aircraft starting on a certain day denoted by $j'$, given the fly history of each aircraft since the last maintenance, such that the $D – day$ maintenance requirement is satisfied. The maintenance is assumed to be performed only at night. In addition, we want to generate the aircraft routes with the minimum difference comparing to the tactical aircraft maintenance routing solutions. As we have discussed in Section 2.3.3, many models for AMRP, e.g., the flight string model, can be extended to model the TAP because TAP and AMRP are closely related. In this section, we demonstrate that the proposed RWLNM can also be extended to model the TAP as well.

To facilitate our discussion, we extend the WLN to model the operational TAP as shown in Fig. 4. In particular, we construct a set of $K$ additional nodes, denoted by $N_k$, each representing an aircraft. We also construct a set of all possible LOFs, each representing a feasible LOF for an aircraft on day $j'$. If an aircraft $k$ has flown $i$ days (not including the current day) without a maintenance, we then connect node $n_k \in N_k$ to a set of end-day nodes of day $j'$ in the $S_{j-1}$ subnetwork with the
constructed LOF arcs. For example as shown in Fig. 4, at the beginning of Thursday, we have three aircraft A, B, and C. Aircraft A has flown two days, aircraft B has flown one day, and aircraft C is freshly maintained on Wednesday night. We then connect node $n_A$ to the end-day nodes of Thursday in the $S_2$ subnetwork starting on Tuesday with the set of LOFs arcs for aircraft A. Similarly, we connect node $n_B$ to the end-day nodes of Thursday in the $S_3$ subnetwork starting with the LOF arcs for aircraft B, and connect node $n_C$ to the end-day nodes of Thursday in $S_4$ subnetwork. Based on the extended WLN, we propose the following TAP model (TAPM):

**Additional Sets, Elements, and Constants**

- $f^*$: the first day in the planning horizon.
- $F_j$: the set of flights to be scheduled on day $j$.
- $N_k$: the set of aircraft nodes, indexed by $n_k$.
- $L_k$: the set of feasible LOFs for aircraft $k$ on day $j$.
- $\hat{N}$: the set of nodes whose inbound arcs contain any element from $\bigcup_{k \in \{1,...,K\}} L_k$ in the extended network.

**Additional Variables**

- $u_k$: is a binary variable such that $u_k = 1$ if LOF $l_k \in L_k$ is selected, and 0 otherwise.

The TAPM formulation is stated as follows:

$$ \text{TAPM} \quad \min \sum_{j \in \{1,...,J\}} \sum_{l_j \in L_j} c_{kj} x_j + \sum_{k \in \{1,...,K\}} \sum_{l_k \in L_k} c_{jk} u_k $$

s.t. $\sum_{l_k \in L_k} u_k = 1 \quad \forall k \in \{1,...,K\}$,

$$ \sum_{k \in \{1,...,K\}} \sum_{l_k \in L_k} \gamma_{l_k}(f) u_k = 1 \quad \forall f \in F_{f^*},$$
The computation of EPD in Eq. (20) together ensure that all the aircraft circulate in WLN. The constraints in Eq. (21) are the binary variable constraints. The remaining of the constraints are the same as in RWLNM presented in Eqs. (10)–(16).

4. Solution methods: heuristic and column generation

Based on our initial analysis, it was found that for a schedule with less than 100 flights per day (approximately 700 flights per week) it is still computationally possible to enumerate all possible LOFs and use commercial MIP solvers such as CPLEX to solve the RWLNM. For large RWAMRP instances, the number of possible LOFs is still too large for complete enumeration. In real life schedules, large commercial airlines usually operate more than 100 flights per fleet per day; therefore, complete enumeration of LOFs is intractable and the RWLNM becomes too large to solve in a reasonable time. In addition, the computation of EPD $C_l$ alone takes almost 50% of the total computational effort in enumerating LOFs because the EPD of an LOF is highly non-linear and its calculation cannot be decomposed. In this paper, we develop a new column generation framework to efficiently construct good LOFs with a small EPD based on specific properties of the EPD and prove the optimality of found solutions to the RWLNM.

4.1. Best allocation of buffer time

The number of variables in RWLNM is governed by the number of LOFs. To solve the RWLNM efficiently, instead of enumerating all the possible LOFs, we can exclude the LOFs that are likely to possess unnecessary long EPDs in the model. Therefore, when generating LOFs, we want to arrange the flights such that the buffer times between flights are allocated optimally or close to optimal, i.e., the buffer times can then absorb the NPD as much as possible. Hence, it is logical to find out the optimal allocation of buffer time as a guideline to generate LOFs. We notice the optimal buffer arrangement is mainly affected by two factors: total buffer time and the number of flights in an LOF. Therefore, we try to find out the optimal buffer arrangement for all reasonable combinations of total buffer time and the number of flights in the LOF. In particular, we compute the EPD of LOF for all possible combination of the buffer times, and record the best buffer time allocation with the minimum EPD. The detailed pseudocode for computing optimal buffer arrangement is shown in Fig. 5. In our preliminary study, we also find out that the optimal buffer arrangement is not sensitive to different historical NPD for different fleets, as long as all the flights share the same NPD distribution when computing the EPD of LOF. Finally, it is worth mentioning that one can always obtain more specific non-propagated delay PMFs (e.g., the PMF associated with airports, departure and arrival time blocks, or even flight numbers) for more accurate optimal buffer allocation, and the proposed method is still valid. However, the detailed analysis of historical non-propagated delay PMFs is beyond the scope of this paper.

The optimal buffer arrangement is shown in Table 3. The first column presents the total buffer time of the LOF in minutes, and the first row indicates the number of flights in the LOF. For each combination of the total buffer time and the number of flights, we present the optimal buffer arrangement and the corresponding EPD. For example, if the total buffer time of an LOF is 2 h and the number of flights in the LOF is 4, from Table 3 we know that the optimal arrangement of buffer time is 36–50–34, and the EPD of this optimal arrangement is approximately 8.5 min. If we generate an LOF with buffer 55–10–45 (notice the sum of the buffer time is also 2 h), by comparing this with the optimal buffer 36–50–34, we know that this new LOF will generate much larger EPD and is less attractive for the proposed model. On the other hand, if we generate a new LOF with buffer 32–55–33, we know this buffer arrangement is similar to the optimal buffer arrangement and with very small EPD, and we will definitely include this LOF in the mathematical model.

$$\sum_{k \in \{1, \ldots, K\}} \sum_{l \in L_k} y_{kl} u_k = \sum_{m \in M, d \in \{1, \ldots, J\}} \sum_{l \in L_k} \beta_{m,d} Z_{md} + y_{g_n} \quad \forall n \in \tilde{N},$$

(20)

Constraints in (10)–(16),

$$u_k \in \{0, 1\} \quad \forall k \in \{1, \ldots, K\}, \forall l \in L_k.$$
Best Buffer Allocation

**Input:**
- Predefined constant $n$: The number of flights $n$ in the LOF
- $\text{Buf} = 30$: Initial buffer time
- $\text{Step} = 30$: Step size for increasing total buffer time
- $\text{BufUpperBound} = 480$: Upper bound for total buffer time

1. $\text{FOR Buf} \leq \text{BufUpperBound}$
2. $\text{BestDelay} = \infty$
3. $\text{FOR All combination of } b_{f_1, f_2}, b_{f_3, f_4}, \ldots, b_{f_{n-1}, f_n}, \text{such that } \sum_{i=1}^{n-1} b_{f_i, f_{i+1}} = \text{Buf}$
4. $\text{CurrentDelay} \leftarrow \text{Compute EPD using Eqs. (5)-(8)}$
5. $\text{IF BestDelay} < \text{CurrentDelay}$
6. $\text{BestDelay} \leftarrow \text{CurrentDelay}$
7. $\text{Record BestDelay Configuration}$
8. $\text{END IF}$
9. $\text{Output BestDelay Configuration}$
10. $\text{END FOR}$
11. $\text{Buf} \leftarrow \text{Buf} + \text{Step}$
12. $\text{END FOR}$

**Fig. 5.** Pseudo-code for computing the best allocation of buffer time.

### Table 3
Optimal buffer allocation table with the minimal propagated delay.

<table>
<thead>
<tr>
<th>Total buffer time (min)</th>
<th>3 Flights</th>
<th>4 Flights</th>
<th>5 Flights</th>
<th>6 Flights</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Optimal buffer allocation</td>
<td>Minimal EPD</td>
<td>Optimal buffer allocation</td>
<td>Minimal EPD</td>
</tr>
<tr>
<td>270</td>
<td>150-150</td>
<td>0.184</td>
<td>110-112-108</td>
<td>1.353</td>
</tr>
<tr>
<td>300</td>
<td>166-164</td>
<td>0.107</td>
<td>120-122-118</td>
<td>1.059</td>
</tr>
<tr>
<td>330</td>
<td>180-180</td>
<td>0.055</td>
<td>130-130-130</td>
<td>0.825</td>
</tr>
<tr>
<td>360</td>
<td>210-210</td>
<td>0.022</td>
<td>140-142-138</td>
<td>0.637</td>
</tr>
<tr>
<td>390</td>
<td>225-225</td>
<td>0.000</td>
<td>150-150-150</td>
<td>0.485</td>
</tr>
<tr>
<td>420</td>
<td>240-240</td>
<td>0.000</td>
<td>160-162-158</td>
<td>0.364</td>
</tr>
</tbody>
</table>

Also, we observe the followings from Table 3.

**Observation 1:** The optimal buffer time allocation time is approximately symmetric, especially when the total buffer time is long.

Intuitively, if we divide the entire LOF into two connecting parts, one from the beginning flight to the flight in the middle, and the other one is from the middle flight to the last flight in the LOF. Because the gradient of EPD is decreasing with the total buffer time, the symmetric allocation of buffer time will produce the smallest EPD. For example, if we allocate much more buffer time to the first part, the second part will have much less buffer. The reduced EPD (comparing with the symmetric allocation of buffer) from the first part will be less than the increasing EPD from the second part. On the other hand, if we allocate much less buffer time to the first part, the first part will produce much longer EPD than the saving of EPD from the second part. Therefore, the optimal allocation of the buffer time has to be symmetric.

**Observation 2:** The buffer times in the middle of the LOF are always significantly larger than the buffer times towards the beginning and the end.

We might think that the buffer time should be allocated more to the earlier flights than the later flights because the delay of the earlier flights might affect all the later flights. However, this intuition is contrasted by **Observation 2.** On one hand, if we allocate much buffer time to the very beginning of the LOF and the delay of the early flight is not likely to propagate to the later flights, then there will be much less buffer time left for all the later flights and there could be significantly large EPDs from the later flights. On the other hand, if we allocate more buffer time toward the end, there will be significant EPD from the early flights.

**Observation 3:** The buffer times towards the beginning of the LOF are slightly longer than the buffer times towards the end.

**Observation 4:** Given a fixed total buffer time, the EPD increases with the number of the flights in an LOF. Furthermore, the incremental EPD increases with the number of the flights in the LOF. For example, when the total buffer time is 180 min,
the incremental EPD is $6.078 - 2.021 = 4.057$ min if the number of flights increases from three to four; and the incremental EPD is $12.493 - 6.078 = 6.415$ min if the number of flights increases from four to five.

From Observation 4, we know that the incremental EPD increases with the number of flights in an LOF. Therefore, it is not profitable to include the LOFs containing too many flights. Specifically, we are only interested in enumerating good LOFs with less or equal to five flights.

### 4.2. Heuristic approach for LOF generation

We develop a heuristic approach to schedule the flights in an LOF such that they follow the optimal buffer allocation patterns as shown in Table 3. However, it is hard to use the exact optimal buffers when the arrival time and the departure time of the flights have been fixed, especially at spoke stations. Therefore, we allow some flexibility on the buffers when constructing LOFs. Particularly, we first plot the optimal buffer times from Table 3 in Fig. 6, then we plot a shadow area along the optimal time line (as shown in Fig. 6), representing an acceptance region. As long as the buffers of an LOF fall into the shadow area, we accept the constructed LOF. In our computation, the region is set to be within $\pm \frac{\text{total buffer time}}{2}$ minutes of the optimal buffer time. Intuitively, the shorter the total buffer time, the rigid the region. This is mainly because a small difference from the optimal buffer allocation would cause a large increase on the EPD. For example, for a three-flight LOF, if the total buffer time is 30 min, a 10-min difference from the optimal buffer allocation will increase the EPD greatly; whereas if the total buffer time is 180 min, a 10-min difference is still near optimal. Similarly, the larger the number of flights in a LOF, the smaller the region. Our preliminary study also suggests that sometimes it is hard to follow the buffer allocation pattern at spoke stations. Therefore, we calculate the difference between the optimal buffer time and the buffer time at the spoke stations for an LOF, and then evenly allocate the difference to the buffers at the hubs.

### 4.3. Column generation framework

For problems with a very large number of flights to be scheduled, we propose a column generation framework to find good LOFs and to prove the optimality of the result. In particular, we utilize two lower bounds for EPD to prune a large number of unprofitable LOFs, and efficiently generate only potentially good LOFs.

#### 4.3.1. Pricing operation:

Let $\theta_l$ be the dual variable associated with constraints in Eq. (10), $\eta_l$ be the dual variable associated with constraints in Eq. (11), and $\xi$ be the negative dual variable associated with the constraint in Eq. (13). The reduced cost $\bar{c}_l$ of LOF $l$ can be written as

$$
\bar{c}_l = c_l - \sum_{f \in F_l} \theta_f + \eta_{n_l} - \eta_{g_l} + \xi \quad \forall l \in L.
$$

The subproblem is thus to find an LOF with minimum $\bar{c}_l$ in any $S_f$ network. Since it is impossible to decompose the EPD in a time-space network or connection network and the accurate computation of $\bar{c}_l$ is very time consuming, we utilize two lower bounds for $\bar{c}_l$, denoted by $\bar{c}_l^1$ and $\bar{c}_l^2$, to speed up the pricing operations. For any LOF $l$, $\bar{c}_l^1$ is define as the approximate EPD computed using Eqs. (1)–(4). $\bar{c}_l^2$ is define as the least possible EPD given LOF $l$’s total buffer time, which can be easily computed from Table 3. $\bar{c}_l$ is then defined by

$$
\bar{c}_l = \max\{\bar{c}_l^1, \bar{c}_l^2\} - \sum_{f \in F_l} \theta_f + \eta_{n_l} - \eta_{g_l} + \xi \quad \forall l \in L.
$$

Clearly, $\bar{c}_l$ is an lower bound of $\bar{c}_l$. If there does not exist an LOF such that $\bar{c}_l < 0$, we know the current LP solution is optimal. If $\bar{c}_l < 0$ for an LOF, we then compute the accurate values of $c_l$ and $\bar{c}_l$. Because the computation of $\bar{c}_l$ is very time consuming, by using lower bound $\bar{c}_l$, we only need to compute $c_l$ for potentially good LOFs and save a large amount of computational effort.

![Fig. 6. The optimal buffer allocation time and the acceptance region for enumerating good LOFs with four flights.](image-url)
In our preliminary study, it is observed that the gap between \(c_l\) and \(c_i\) could be as much as 40%. Therefore, \(c_i\) might be less accurate when the gap between \(c_i\) and \(c_l\) is large. Here, we propose a method to efficiently tighten the approximation of \(c_i\). In particular, we decompose the computation of the EPD for the flight \(f_n\) in an LOF (shown in Eq. (7)) into two parts when \(n > 5\). We first compute EPD for flight \(f_{[n/2]}\) using Eq. (7). Then we deduct the buffer time \(b_{f_{[n/2]}}\) by EPD. Subsequently, we compute the EPD starting from flight \(f_{[n/2]}\) with the updated buffer time \(b_{f_{[n/2]+1}}\). By doing so, we reduce the computation of \(c_i\) from \(T^n\) to \(2 \times T_{[n/2]}\). This new approximation is denoted by \(c_i'\). Intuitively, the computation of \(c_i\) hybridizes the idea of computing \(c_l\) and \(c_i'\), and it can be easily proved that \(c_i \geq c_i' \geq c_l\).

4.3.2. LOF generation procedure:

We develop a two-stage column generation approach to solve the RWLNM. In the first stage, we solve the RWLNM with the approximate cost \(c_i\). Then based on the optimal solution to the first stage problem, we select a set of the LOFs such that the absolute value of the reduced costs are less than a very small positive number \(\epsilon\). These LOFs then serve as initial LOFs for the second stage RWLNM with accurate cost \(c_i\). We then compute the \(c_i\) for these LOFs and then use the column generation method in Section 4.3.1 to solve the second stage problem optimally. The detailed algorithm is described as follows:

**Step 1:** Generate a set of initial LOFs using the heuristic approach in Section 4.2, and solve the LP relaxation of RWLNM with the approximate cost \(c_i\) using the column generation.

**Step 2:** Evaluate the reduced cost of every LOF after the last iteration of the column generation in Step 1. If the absolute value of the reduced cost of an LOF is less than \(\epsilon\), we select it as one of initial LOFs for the second stage RWLNM and compute the accurate cost \(c_i\) using Eqs. (5) and (6).

**Step 3:** Solve the LP relaxation of RWLNM with the accurate cost \(c_i\) using the column generation described in Section 4.3.1.

**Step 4:** Solve the IP version of RWLNM with the set of LOFs generated in Step 2 and Step 3.

The purpose of Step 1 and Step 2 is to obtain an elite set of LOFs, and to compute the \(c_i\). Specifically, in Step 1, we do not need to solve the column generation optimally. Instead, we only need to obtain a good enough feasible solution for the second stage column generation. In Step 2, we select a set of LOFs with small absolute reduced costs from Step 1. In our computation, we set \(\epsilon = 0.01\). In Step 3, we use a breadth first search to find good LOFs. Because the complexity of computing \(c_i\) increases drastically with the number of flights in an LOF, we want to ensure that all LOFs containing a small number of flights have been searched before the search tree grows one more level deeper. In each iteration of the second stage column generation, we stop generating new LOFs if the number of LOFs with good reduced costs has reached 100, and add these new LOFs into the restricted master problems. In Step 4, we use the set of good LOFs generated from Step 2 and Step 3 to obtain an IP solution. If the IP solution is not satisfied, we use branch-and-price to obtain the optimal IP solution.

4.3.3. Variations on the cost of LOFs

In real-life operations, longer delays are much more costly and damaging on subsequent flights in the schedule than the shorter delays. The relationship between the delay and its cost is usually non-linear. Therefore, the airline might want to use a different cost function when planning the aircraft rotation. Hence, it becomes critical if our column generation framework can be used for different objective functions.

In particular, if the cost of an LOF, denoted by \(\tilde{c}_i\), can be computed using the following equations, we can use the proposed method to solve the problem optimally:

\[
\text{COST}_{f_2} = \sum_{t_{j_2}=0}^{T} h((t_{j_1} - b_{f_2})^+) p_{t_{j_1}}
\]

\[
\text{COST}_{f_3} = \sum_{t_{j_2}=0}^{T} \sum_{t_{j_3}=0}^{T} h((t_{j_1} - b_{f_3})^+ + t_{j_2} - b_{f_2})^+) p_{t_{j_1}} p_{t_{j_2}}
\]

\[
 \vdots 
\]

\[
\text{COST}_{f_n} = \sum_{t_{j_{n-1}}=0}^{T} \cdots \sum_{t_{j_2}=0}^{T} \sum_{t_{j_1}=0}^{T} h((\cdots (t_{j_1} - b_{f_2})^+ + \cdots + t_{j_{n-1}} - b_{f_{[n/2]+1}})^+) p_{t_{j_1}} \cdots p_{t_{j_{n-1}}}
\]

\[
\tilde{c}_i = \sum_{l=2}^{n} \text{COST}_{f_l} \quad \forall l \in L.
\]

The Eqs. in (23)–(27) are similar with Eqs. in (5)–(8). Instead of computing the summation of EPD for each flight, we compute the summation of expected value of the cost function \(h\) on propagated delays for all the flights in an LOF.
5. Computational results

In this section, we report empirical results of the proposed models. All test problems were solved using a Dell Precision T7600 workstation with two Intel Xeon E5-2643 CPUs of 3.3 GHz, and 64 GB of memory on a 64-bit Windows 7 platform. Computational times reported in this section were obtained from the computer's internal timing calculations. All the mathematical modeling and algorithms were implemented in C++ language. All LP and MIP problems were solved using a CPLEX callable library version 12.5. All the algorithms and simulations were coded using Microsoft Visual C++ 2010. Parallel computing of eight threads were programmed and implemented when generating LOFs to speed up the computational time.

5.1. Historical data analysis

In this section, we report the analysis of historical data obtained from the Bureau of Transportation Statistics (BTS) (BTS, 2014). We obtained the flight information for all the flights recorded by the BTS from December 2011 to November 2012. The data contains a total of 6,096,432 flights. We selected the top three fleets with the most average propagated delay from a major domestic airline. In Table 4, we present the number of the total flights in the airline fleet, the number of delayed flights, the total delay minutes of non-propagated delay, the total delay minutes of propagated delay, the average non-propagated delay, and the average propagated delay. In this study, a flight was considered a delayed flight although its delay time was less than 15 min. This is because any delay may be able to cause the propagated delay in the real operation. The total non-propagated delay and the propagated delay from all the delayed flights were subsequently computed. The average non-propagated delay and the average propagated delay were computed by dividing total non-propagated delay and total propagated delay by the total number of flights in the fleet, respectively. In Table 4, it is observed that about 17–20% of the flights in these three fleets were delayed flights. The propagated delay was accounted for about 40–45% of the total delay.

Fig. 7 illustrates the probability mass functions (PMFs) of non-propagated delay for the three fleets presented in Table 4. It is interesting to note that the PMFs for all three fleets are quite similar, and over 90% of the area under the three PMFs overlaps. Therefore, to simplify our computation in this study, we constructed a single non-propagated delay PMF by taking an average of the three PMFs. Note that one can always obtain more specific non-propagated delay PMFs (e.g., the PMF associated with airports, departure and arrival time blocks, or even flight numbers) to improve the accuracy of the EPD. However, such detailed analysis of historical non-propagated delay PMFs is beyond the scope of this paper.

We also investigated the number of flights in all LOFs from the airline because the maximum number of flights in LOFs affects the number of all possible LOFs. In Table 5, we show the number of LOFs containing different number of flights from December 2011. It is observed that there is no LOF with more than eight flights, and the LOFs with six or seven flights only contain less than 3% of the total LOFs. Thus, this observation supports our computational approach in generating LOFs with five or fewer flights.

5.2. Realistic test instance construction of RWAMRP

A total of eight realistic test instances were generated and used to benchmark our approaches. These instances were previously used in our previous study Liang and Chaovalitwongse (2013). These instances were constructed based on real life operational aircraft schedules of a domestic airline, as previously mentioned in Section 5.1. The detailed information of eight

Table 4
Top three fleets with the longest average non-propagated delay for a domestic airline.

<table>
<thead>
<tr>
<th>Fleet</th>
<th>Total flights</th>
<th>Delayed flights</th>
<th>NPD (in min)</th>
<th>PD (in min)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Num</td>
<td>%</td>
</tr>
<tr>
<td>A320-1/2</td>
<td>495,064</td>
<td>91,492</td>
<td>18.48</td>
<td>3,388,312</td>
</tr>
<tr>
<td>B757</td>
<td>473,657</td>
<td>94,522</td>
<td>19.96</td>
<td>3,589,488</td>
</tr>
<tr>
<td>A319</td>
<td>284,673</td>
<td>49,254</td>
<td>17.30</td>
<td>1,877,498</td>
</tr>
</tbody>
</table>

Fig. 7. The probability density function for the non-propagated delay of three fleets from a domestic airline.
test cases are shown in Table 6. The first six test instances were constructed from six individual fleets whereas the last two instances were constructed by combining multiple fleets to create large instances.

The following procedure was carried out to construct the first six test instances. First, we selected six representative fleets and extracted the corresponding airline flights within a particular week of the published schedule. The fleet size information was obtained from the airline website. However, because there was no information available on the maintenance stations, airports with the numbers of departure/arrival flights greater than a threshold were assumed to be maintenance stations. This threshold was fine-tuned so that the total number of maintenance stations was minimized while maintaining the feasibility of the schedule. For instance, the threshold for Boeing 757-200 fleet was set to 30, i.e., an airport was assumed to be a maintenance station if the pairs of departure/arrival flights at the airport were greater than 30. Also, the maintenance capacity of a station is computed by:

\[ Q_m = \frac{Q_{mj}}{\text{number of arrivals at } m \text{ within a week/maintenance threshold}}, \exists m \in M, \forall j \in \{1, 2, \ldots, 7\}. \]

For example, there are 241 flights departing from ORD weekly for Boeing 757–200 fleet, therefore, we set the maintenance capacity at ORD to be 8 daily. To create large test instances, we integrated the schedules of multiple fleets into a single schedule of one combined fleet. Specifically, the schedules of AIR-320 and 737-800 fleets were combined to create the SIM-001 instance. The schedules of 757-200, AIR-320, and 737-800 fleets were combined to create the SIM-002 instance. It is interesting to note that the last test instance, SIM-002, is about the size of the world’s largest fleet, Southwest Airline Boeing 737-700 (350 aircrafts). For all the test cases, the maximum number of days between two consecutive maintenances is assumed to be four days, following a realistic estimation of airline regulations.

5.3. RWAMRP solutions without column generation

In this section, we report the computational results of RWLNM for all eight test instances without using column generation. To solve the WRLNM in a reasonable time, we enumerated all possible LOFs that contained only five or fewer flights. Parallel processing with eight threads was used when generating LOFs. In Table 7, Cols, Rows and Non0s indicate the number of columns, the number of rows, the number of non-zero coefficients in the problem, respectively. LOFs and LOF Time indicate the number of LOFs and the LOF generation time, which includes the time of enumerating all LOFs and the time of computing EPDs. LP, IP, LP – IP Gap, and IP Time indicate the LP objective value, the IP objective value, the LP-IP gap, and the model-solving time, respectively. The LP – IP Gap was computed by (IP – LP)/IP.

<table>
<thead>
<tr>
<th>Test cases</th>
<th>Flights</th>
<th>Fleet size</th>
<th>Redeyes</th>
<th>Airports</th>
<th>Maint stations</th>
</tr>
</thead>
<tbody>
<tr>
<td>757-300</td>
<td>348</td>
<td>22</td>
<td>14</td>
<td>18</td>
<td>2</td>
</tr>
<tr>
<td>737-500</td>
<td>658</td>
<td>35</td>
<td>0</td>
<td>34</td>
<td>3</td>
</tr>
<tr>
<td>CRJ-700</td>
<td>972</td>
<td>55</td>
<td>0</td>
<td>40</td>
<td>6</td>
</tr>
<tr>
<td>757-200</td>
<td>1428</td>
<td>88</td>
<td>54</td>
<td>31</td>
<td>6</td>
</tr>
<tr>
<td>AIR-320</td>
<td>2080</td>
<td>123</td>
<td>38</td>
<td>62</td>
<td>7</td>
</tr>
<tr>
<td>737-800</td>
<td>2240</td>
<td>122</td>
<td>54</td>
<td>84</td>
<td>12</td>
</tr>
<tr>
<td>SIM-001</td>
<td>4342</td>
<td>245</td>
<td>92</td>
<td>107</td>
<td>28</td>
</tr>
<tr>
<td>SIM-002</td>
<td>6072</td>
<td>333</td>
<td>156</td>
<td>109</td>
<td>30</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Test cases</th>
<th>Cols</th>
<th>Rows</th>
<th>Non0</th>
<th>LOFs</th>
<th>LOF Time</th>
<th>LP</th>
<th>IP</th>
<th>LP-IP Gap</th>
<th>IP Time</th>
<th>Total time</th>
</tr>
</thead>
<tbody>
<tr>
<td>757-300</td>
<td>9655</td>
<td>1427</td>
<td>2232</td>
<td>1</td>
<td>212</td>
<td>212</td>
<td>0.00%</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>737-500</td>
<td>75,102</td>
<td>2815</td>
<td>393,794</td>
<td>18,282</td>
<td>13</td>
<td>782</td>
<td>782</td>
<td>0.00%</td>
<td>4</td>
<td>18</td>
</tr>
<tr>
<td>CRJ-700</td>
<td>184,221</td>
<td>3990</td>
<td>1,037,173</td>
<td>45,357</td>
<td>88</td>
<td>1239</td>
<td>1239</td>
<td>0.00%</td>
<td>12</td>
<td>101</td>
</tr>
<tr>
<td>757-200</td>
<td>269,661</td>
<td>3207</td>
<td>1,460,253</td>
<td>71,260</td>
<td>57</td>
<td>910</td>
<td>910</td>
<td>0.00%</td>
<td>47</td>
<td>104</td>
</tr>
<tr>
<td>AIR-320</td>
<td>602,000</td>
<td>5938</td>
<td>3,438,081</td>
<td>155,180</td>
<td>220</td>
<td>1489</td>
<td>1497</td>
<td>0.07%</td>
<td>139</td>
<td>360</td>
</tr>
<tr>
<td>737-800</td>
<td>1,383,632</td>
<td>7085</td>
<td>7,948,309</td>
<td>354,369</td>
<td>434</td>
<td>2446</td>
<td>2446</td>
<td>0.00%</td>
<td>474</td>
<td>910</td>
</tr>
<tr>
<td>SIM-001</td>
<td>3,358,298</td>
<td>10,755</td>
<td>19,783,822</td>
<td>868,949</td>
<td>1412</td>
<td>2798</td>
<td>2798</td>
<td>0.00%</td>
<td>3493</td>
<td>4910</td>
</tr>
<tr>
<td>SIM-002</td>
<td>11,728,242</td>
<td>12,611</td>
<td>71,869,753</td>
<td>3,141,711</td>
<td>6176</td>
<td>FAIL</td>
<td>FAIL</td>
<td>–</td>
<td>12,871</td>
<td>19,051</td>
</tr>
</tbody>
</table>
From Table 7, we obtained the optimal solutions (with only LOFs with five or less flights) for seven out of eight test instances in less than two hours. It is important to note that these solutions might not be truly optimal if we allow more flights to be included in an LOF. In fact, this is the case (not globally optimal) as shown in the next subsection. From the table, the LP bounds provided by the RWLNM were very tight as zero LP-IP gaps were obtained in six out of first seven test instances and the only non-zero gap was less than 0.1%. However, for the last instance, SIM-002, CPLEX failed to provide a feasible LP and IP solution after more than three hours of run time even though it was known during our test instance reconstruction that these exists an optimal solution in this instance. The reason that CPLEX failed was mainly due to the drastically increased number of LOFs (over 3 million), which resulted into more than 11 million binary variables in the RWLNM. It is also worth noting that, for the last test case, SIM-002, it took more than 100 min to compute the EPDs and the computation of the accurate cost $c_l$ for all LOFs with six flights required more than 24 h of run time.

Table 8 shows a comparison of computational results using three different methods to calculate EPDs. In the table, LOFs indicates the number of LOFs enumerated. The computational time (in seconds) for enumerating LOFs and computing the costs, the IP objective value, and IP computational time (in seconds) for different costs $c_l$, $c_i$, and $c_o$ were also reported. In addition, the gaps between IP objective values, $\text{GAP} = \frac{\text{ACCIP} - \text{APPIP}}{\text{ACCIP}}$, were provided in the table.

It can be seen that the computational time of EPDs was greatly reduced if the approximation costs $c_l$, $c_i$, or $c_o$ was used. For example, the EPD computational time in SIM-002 was reduced from more than 100 min to less than 10 min by using $c_o$ and to about 15 min by using $c_l$. On the other hand, the IP computational times were not significantly affected by the costs. Nevertheless, the GAP on $c_l$ suggests that the approximate method underestimated the resulting EPDs by an average of 7.16%, ranging from 1.89% to over 14% whereas the GAP on $c_l$ shows that $c_l$ provided very accurate (less than 2%) approximation on the resulting EPDs. It is also observed that the more flights in the resulting LOFs, the larger gaps to the accurate optimal solutions.

### 5.4. RWAMRP solutions with column generation

When solving the RWLNM, we used the approximation $c_l$ as the cost for the first stage column generation. We solved the first stage LP optimally until there was no LOF with a good reduced cost, and subsequently solved the second stage column generation with a more accurate cost $c_l$. Note that $c_l$ was also used to speed up the pricing operation as described in Section 4.3.1. The second stage column generation continued until there was no LOF with a good reduced cost. Then we solved an IP program with the generated LOFs to obtain an integral solution. In this study, we used parallel processing of eight threads when generating LOFs and computing their costs.

Table 9 shows computational results of our two-stage column generation approach. In particular, we report the LP objective value and computational time of the first stage column generation, the initial number of LOFs, the LOFs generated by

### Table 8
Comparison of three different methods to compute the EPD of an LOF.

<table>
<thead>
<tr>
<th>Test cases</th>
<th>LOFs</th>
<th>Accurate LOF cost</th>
<th>Approximate LOF cost</th>
<th>Approximate LOF cost</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>$c_l$</td>
<td>$c_i$</td>
<td>$c_o$</td>
</tr>
<tr>
<td>757-300</td>
<td>2232</td>
<td>1</td>
<td>212</td>
<td>1</td>
</tr>
<tr>
<td>737-500</td>
<td>18.282</td>
<td>13</td>
<td>782</td>
<td>4</td>
</tr>
<tr>
<td>CRJ-700</td>
<td>45.357</td>
<td>88</td>
<td>1239</td>
<td>12</td>
</tr>
<tr>
<td>757-200</td>
<td>71.260</td>
<td>57</td>
<td>910</td>
<td>47</td>
</tr>
<tr>
<td>AIR-320</td>
<td>155.180</td>
<td>220</td>
<td>1497</td>
<td>139</td>
</tr>
<tr>
<td>757-500</td>
<td>354.369</td>
<td>434</td>
<td>2446</td>
<td>474</td>
</tr>
<tr>
<td>SIM-001</td>
<td>868,949</td>
<td>1412</td>
<td>2798</td>
<td>3493</td>
</tr>
<tr>
<td>SIM-002</td>
<td>3,141,711</td>
<td>6176</td>
<td>FAIL</td>
<td>12,871</td>
</tr>
</tbody>
</table>

### Table 9
Computational results of RWLNM using the column generation approach on all eight test instances.

<table>
<thead>
<tr>
<th>Test cases</th>
<th>Col Gen with $c_l$</th>
<th>Col Gen with $c_l$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>LP Val</td>
<td>Col-Gen Time</td>
</tr>
<tr>
<td>757-300</td>
<td>212</td>
<td>2</td>
</tr>
<tr>
<td>737-500</td>
<td>778</td>
<td>6</td>
</tr>
<tr>
<td>CRJ-700</td>
<td>1238</td>
<td>19</td>
</tr>
<tr>
<td>757-200</td>
<td>910</td>
<td>17</td>
</tr>
<tr>
<td>AIR-320</td>
<td>1489</td>
<td>54</td>
</tr>
<tr>
<td>737-800</td>
<td>2405</td>
<td>137</td>
</tr>
<tr>
<td>SIM-001</td>
<td>2760</td>
<td>545</td>
</tr>
<tr>
<td>SIM-002</td>
<td>3079</td>
<td>2080</td>
</tr>
</tbody>
</table>
column generation, the number of iterations, the objective value, and the computational time of the second stage column generation. We also report the solution value and the computational time of the final IP. From the table, the optimal LP solutions to all test instances were obtained within three hours. The optimal IP solutions were obtained in five out of eight test instances, and the integer solutions within 0.6% gap to the optimal solutions were obtained in the rest three test instances. The results demonstrate that the first stage column generation efficiently provided very good approximations (within 2%) of the optimal solutions, making the initial solutions of the second stage column generation very tight as compared to the optimal LP solution (within 0.6%). Comparing Tables 7 and 9, the two-stage column generation approach clearly outperformed the basic enumeration approach (without column generation) as it provided far better solution quality and shorter computational time. Specifically, the column generation approach obtained better IP solutions in three instances (737-800, SIM-001, and SIM-002) and comparable IP solutions in other instances while the computational time of the column generation approach was much shorter than that of the basic enumeration method, especially in the last instance.

As the computational time of the column generation approach was heavily dependent on the computational time of $c_l$, especially for a number of flights in $l$, we kept statistics of the number of long LOFs (with five, six, or seven flights) with computed $c_l$ as shown in Table 10. From the table, it is observed that a very small portion of all possible LOFs required the computation of $c_l$ when the number of flights in the LOFs was large. In the two large test instances, SIM-001 and SIM-002, we only needed to compute less than 0.5% of the long LOFs. We also notice that the LOFs with seven flights were rarely considered in the column generation process. In six out of eight test instances, we did not have to compute any accurate $c_l$ for LOFs with seven flights whereas we computed $c_l$ for less than 10 LOFs with seven flights in the other two instances.

5.5. Sensitivity analysis on the cost of LOF

There are two main purposes for the sensitivity analysis on the cost of LOF: First, we want to investigate if one can reduce the propagated delay when changing the cost $c_l$ (as proposed in the literature) to $c_l$; Second, we want to investigate if one can improve the aircraft routing solution quality by using different objective functions in RWAMRP. Therefore, we conducted a sensitivity analysis by changing the cost of LOF to $c_l$ and $c_l$. Here, we define cost function $h$ in Eqs. (24)–(27) as follows.

$$h(x) = \begin{cases} \frac{16}{x} & \text{if } x \leq 15; \\ x \times (\frac{16}{x}) + 1 & \text{if } x \geq 16. \end{cases}$$

(28)

By using cost function $h$, we penalize less on the delays that are shorter than or equal to 15 min, and penalize more on the delays that are longer. We postulate that the proposed cost $c_l$, defined in Eqs. (24)–(27), may improve the resulting schedule, especially when very long propagated delays are encountered. To carry out the analysis, we solved the RWLN with costs $c_l$ and $c_l$ to obtain an optimal weekly schedule for each value of the costs. We subsequently simulated 500 consecutive weekly realizations of a random NPD for every flight that follows the NPD distribution, as discussed in Section 5.1, which equated to a ten-year duration of flight scheduling. We then assessed the performance of individual optimal schedules including an average propagated delay (in minutes) and an average number of propagated delayed flights per weekly realization as well as the total number of flights with different propagated delay durations over the entire simulation. The results are reported in Table 11. From the table, $c_l$ is seen to provide the worst simulation results on an average propagated delay duration for all eight test cases. An average difference of the propagated delays between the $c_l$ schedules and the best schedules is around 2.47%, ranging from 1.5% to 4% for the eight test cases. The other two costs $c_l$ and $c_l$ provided similar results in terms of an average propagated delay and the number of flights with propagated delay. However, the schedule obtained from $c_l$ provided a smaller number of flights with very long propagated delays (over three or four hours).

5.6. Operational tail assignment solutions

In this section, we present the computational results of the TAPM. In particular, we created two scenarios for each of the eight schedules. For the first and second scenarios, we set 10% and 20% of the aircrafts to be critical, respectively. For each

<table>
<thead>
<tr>
<th>Test cases</th>
<th>Complete enumeration</th>
<th>Column generation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Five flights</td>
<td>Six flights</td>
</tr>
<tr>
<td>737-300</td>
<td>60</td>
<td>0</td>
</tr>
<tr>
<td>737-500</td>
<td>1541</td>
<td>1840</td>
</tr>
<tr>
<td>CRJ-700</td>
<td>7518</td>
<td>19,138</td>
</tr>
<tr>
<td>757-200</td>
<td>8251</td>
<td>1171</td>
</tr>
<tr>
<td>AIR-320</td>
<td>33,709</td>
<td>14,315</td>
</tr>
<tr>
<td>737-800</td>
<td>35,252</td>
<td>75,459</td>
</tr>
<tr>
<td>SIM-001</td>
<td>203,277</td>
<td>136,528</td>
</tr>
<tr>
<td>SIM-002</td>
<td>1,114,720</td>
<td>537,322</td>
</tr>
</tbody>
</table>
In Table 12, we report the number of LOFs generated for the critical aircrafts, the objective function values and the solution times of the TAPM solutions. As opposed to the RWLNM, the TAPM contained additional constraints on the critical aircrafts. We also report the increment of the EPD from the RWLNM to the TAPM in the objective function value. From the table, all the test cases were optimally solved within 40 min. We only needed to generate a very small number of new LOFs for the

critical aircraft, we assigned a randomly generated time between four hours to six hours as the remaining flight time before the next maintenance. When solving the TAPM, we kept all the LOFs generated during the column generation in the model. We also enumerated the complete set of LOFs for the critical aircrafts. Note that the departure airports of the critical aircrafts were fixed, the arrival airports of the critical aircraft had to be maintenance stations, and the upper bounds on the flight time of the critical aircraft were decided. Therefore, the complete number of the LOFs for critical aircraft was not very large.

Table 11
The simulation statistics with LOF costs $c_1$, $c_2$, and $c_3$.

<table>
<thead>
<tr>
<th>Test cases</th>
<th>Cost</th>
<th>Average propagated delay per replication</th>
<th>Num of propagated delayed flights per replication</th>
<th>Number of flights with different propagated delay durations over 500 simulation replications</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$c_1$</td>
<td>216.22</td>
<td>4.88</td>
<td>770</td>
</tr>
<tr>
<td>757-300</td>
<td>$c_2$</td>
<td>208.00</td>
<td>4.85</td>
<td>776</td>
</tr>
<tr>
<td></td>
<td>$c_3$</td>
<td>213.65</td>
<td>4.82</td>
<td>724</td>
</tr>
<tr>
<td>737-500</td>
<td>$c_1$</td>
<td>807.57</td>
<td>20.03</td>
<td>3794</td>
</tr>
<tr>
<td></td>
<td>$c_2$</td>
<td>791.94</td>
<td>20.20</td>
<td>3935</td>
</tr>
<tr>
<td></td>
<td>$c_3$</td>
<td>788.93</td>
<td>20.11</td>
<td>3810</td>
</tr>
<tr>
<td>CRJ-700</td>
<td>$c_1$</td>
<td>1244.17</td>
<td>34.71</td>
<td>6750</td>
</tr>
<tr>
<td></td>
<td>$c_2$</td>
<td>1219.59</td>
<td>34.79</td>
<td>6892</td>
</tr>
<tr>
<td></td>
<td>$c_3$</td>
<td>1202.20</td>
<td>34.42</td>
<td>6858</td>
</tr>
<tr>
<td>757-200</td>
<td>$c_1$</td>
<td>929.42</td>
<td>22.05</td>
<td>3700</td>
</tr>
<tr>
<td></td>
<td>$c_2$</td>
<td>912.56</td>
<td>21.60</td>
<td>3553</td>
</tr>
<tr>
<td></td>
<td>$c_3$</td>
<td>920.42</td>
<td>22.20</td>
<td>3769</td>
</tr>
<tr>
<td>737-800</td>
<td>$c_1$</td>
<td>1528.82</td>
<td>38.80</td>
<td>7269</td>
</tr>
<tr>
<td></td>
<td>$c_2$</td>
<td>1505.77</td>
<td>38.76</td>
<td>7486</td>
</tr>
<tr>
<td></td>
<td>$c_3$</td>
<td>1522.26</td>
<td>38.59</td>
<td>7335</td>
</tr>
<tr>
<td>AIR-320</td>
<td>$c_1$</td>
<td>2590.61</td>
<td>58.89</td>
<td>9807</td>
</tr>
<tr>
<td></td>
<td>$c_2$</td>
<td>2451.59</td>
<td>57.97</td>
<td>9760</td>
</tr>
<tr>
<td></td>
<td>$c_3$</td>
<td>2457.21</td>
<td>57.94</td>
<td>9792</td>
</tr>
<tr>
<td>737-800</td>
<td>$c_1$</td>
<td>2864.10</td>
<td>68.94</td>
<td>11,907</td>
</tr>
<tr>
<td></td>
<td>$c_2$</td>
<td>2815.79</td>
<td>68.68</td>
<td>11,968</td>
</tr>
<tr>
<td></td>
<td>$c_3$</td>
<td>2854.63</td>
<td>69.00</td>
<td>11,965</td>
</tr>
<tr>
<td>SIM-001</td>
<td>$c_1$</td>
<td>3277.60</td>
<td>77.35</td>
<td>12,726</td>
</tr>
<tr>
<td></td>
<td>$c_2$</td>
<td>3235.32</td>
<td>76.99</td>
<td>12,812</td>
</tr>
<tr>
<td></td>
<td>$c_3$</td>
<td>3198.74</td>
<td>76.64</td>
<td>12,770</td>
</tr>
<tr>
<td>SIM-002</td>
<td>$c_1$</td>
<td>2509.61</td>
<td>58.89</td>
<td>9807</td>
</tr>
<tr>
<td></td>
<td>$c_2$</td>
<td>2451.59</td>
<td>57.97</td>
<td>9760</td>
</tr>
<tr>
<td></td>
<td>$c_3$</td>
<td>2457.21</td>
<td>57.94</td>
<td>9792</td>
</tr>
<tr>
<td>737-800</td>
<td>$c_1$</td>
<td>2864.10</td>
<td>68.94</td>
<td>11,907</td>
</tr>
<tr>
<td></td>
<td>$c_2$</td>
<td>2815.79</td>
<td>68.68</td>
<td>11,968</td>
</tr>
<tr>
<td></td>
<td>$c_3$</td>
<td>2854.63</td>
<td>69.00</td>
<td>11,965</td>
</tr>
<tr>
<td>SIM-001</td>
<td>$c_1$</td>
<td>3277.60</td>
<td>77.35</td>
<td>12,726</td>
</tr>
<tr>
<td></td>
<td>$c_2$</td>
<td>3235.32</td>
<td>76.99</td>
<td>12,812</td>
</tr>
<tr>
<td></td>
<td>$c_3$</td>
<td>3198.74</td>
<td>76.64</td>
<td>12,770</td>
</tr>
</tbody>
</table>

The numbers in bold represent the best simulation results.
critical aircrafts. When the number of the critical aircrafts was 10\% of the fleet size, the EPD solution did not change much from the original RWAMRP solution. However, when the number of the critical aircrafts was 20\% of the fleet size, the solution EPDs increased by an average of 0.4\%. We also tried to increase the number of critical aircrafts to 25\% of the fleet size but no feasible solution was found in AIR-320.

6. Conclusion

In this paper, we present a new robust LOF network model for the RWAMRP. The proposed model hybridizes the flight string model (Barnhart et al., 1998; Lan et al., 2006) and the aircraft rotation tour model (Liang et al., 2011; Liang and Chaovalitwongse, 2013). To solve the model efficiently, we proposed a two-stage column generation to generate good aircraft LOFs, and to prove the optimality of the solution. We also reveal that the traditional way of computing EPD underestimates the actual EPD. Our computational study shows that the underestimation can be as much as 40\% for some LOFs. However, the accurate computation of EPD is very time-consuming, especially when the number of flights in the LOF is large. Therefore, we develop three lower bounds on the EPD and utilize these bounds to speed up the pricing subproblem of the column generation. The computational results show that the model generate very tight LP relaxation, and the column generation can obtain optimal LP relaxation solutions and near optimal IP solutions (within 0.6\% optimal) in less than three hours for very large test cases with more than 6000 flights per week. We also analysis the effect of the proposed model using simulation. Our simulation results show that the proposed model generates better solutions than the traditional methods for all eight test cases efficiently. We also extend the RWLINM for the operational tail assignment model, and the computational results are presented. Finally, it is worth mentioning that the proposed model can be generalized to be integrated with other airline operational planning problems.

Acknowledgements

The first author is supported by National Science Foundation of China [Grant 71422003]. The third author is supported by National Science Foundation of China [Grant 71125004].

Appendix A. Proof of $EPD_f \geq EPD'_f$

We use induction to prove that $EPD_{f_i} \geq EPD'_f$ for all $i \in \{2, \ldots, n\}$. When $i = 2$, $EPD'_f = EPD_f$. Assuming $EPD_{f_i} \geq EPD'_f$, we need to show that $EPD_{f_{i+1}} \geq EPD'_f$.

To prove that, we need two equations as follows:

$$a^p = (ap)^+,$$

$$\sum_i a_i^+ \geq (\sum_i a_i)^+,$$

where $a$ and $p$ can be any real numbers. We also have

$$EPD_f = \sum_{i_{f_i} = 0}^{T} \cdots \sum_{i_{f_{i-1}} = 0}^{T} \left( (\cdots (t_{f_{i}} - b_{f_{i-2}f_{2}})^+ + \cdots + t_{f_{i-1}} - b_{f_{i-1}f_{1}})^+ + t_{f_{i-1}} - b_{f_{i-1}f_{1}} \right) p_{i_{f_{i}}} \cdots p_{i_{f_{1}}} \cdots p_{i_{f_{1}}},$$

$$EPD'_{f_{i+1}} = \sum_{i_{f_{i}}}^{T} \left( EPD_f + t_{f_{i}} - b_{f_{i+1}f_{i+1}} \right)^+ p_{i_{f_{i}}}$$

We can rewrite $EPD_{f_{i+1}}$ as follows:

$$\begin{align*}
EPD_{f_{i+1}} &= \sum_{i_{f_{i}}}^{T} \sum_{i_{f_{i-1}}}^{T} \sum_{i_{f_{i-2}}}^{T} \left( (\cdots (t_{f_{i}} - b_{f_{i-2}f_{2}})^+ + \cdots + t_{f_{i-1}} - b_{f_{i-1}f_{1}})^+ + t_{f_{i-1}} - b_{f_{i-1}f_{1}} \right)^+ p_{i_{f_{i}}} \cdots p_{i_{f_{1}}} \cdots p_{i_{f_{1}}} \\
&= \sum_{i_{f_{i}}}^{T} \sum_{i_{f_{i-1}}}^{T} \sum_{i_{f_{i-2}}}^{T} \left( (\cdots (t_{f_{i}} - b_{f_{i-2}f_{2}})^+ + \cdots + t_{f_{i-1}} - b_{f_{i-1}f_{1}})^+ p_{i_{f_{i}}} \cdots p_{i_{f_{1}}} \cdots p_{i_{f_{1}}} \\
&\quad + (t_{f_{i}} - b_{f_{i+1}f_{i+1}}) p_{i_{f_{i}}} \cdots p_{i_{f_{1}}} \cdots p_{i_{f_{1}}} \right)^+ \ \text{using Eq. (A.1)} \\
&\geq \sum_{i_{f_{i}}}^{T} \left( \sum_{i_{f_{i-1}}}^{T} \cdots \sum_{i_{f_{i-2}}}^{T} \left( (\cdots (t_{f_{i}} - b_{f_{i-2}f_{2}})^+ + \cdots + t_{f_{i-1}} - b_{f_{i-1}f_{1}})^+ p_{i_{f_{i}}} \cdots p_{i_{f_{1}}} \cdots p_{i_{f_{1}}} \\
&\quad + (t_{f_{i}} - b_{f_{i+1}f_{i+1}}) p_{i_{f_{i}}} \cdots p_{i_{f_{1}}} \cdots p_{i_{f_{1}}} \right) \right)^+ \ \text{using Eq. (A.2)}
\end{align*}$$
\[
\sum_{t_{i-1}} T \left( \sum_{t_i} \cdots \sum_{t_{i-1}} (t_{f_j} - b_{f_j}) \right) + \sum_{t_i} \cdots \sum_{t_{i-1}} (t_{f_j} - b_{f_j}) \right) \cdot p_t_i \cdots p_{t_{i-1}} \cdot p_{t_i} \\
+ \sum_{t_i} \cdots \sum_{t_{i-1}} (t_{f_j} - b_{f_j}) \cdot p_t_i \cdots p_{t_{i-1}} \cdot p_{t_i} \\
= \sum_{t_i} \left( EPD_f + (t_{f_i} - b_{f_i}) \right) + \text{using Eq. (A.3)} \\
\geq \sum_{t_i} \left( EPD_f' + (t_{f_i} - b_{f_i}) \right) + \text{p}_{t_i} \\
= EPD_f'_{i-1} 
\]

References
