A passenger demand model for airline flight scheduling and fleet routing

Shangyao Yan*, Chich-Hwang Tseng

Department of Civil Engineering, National Central University, Chungli 32054, Taiwan

Received 1 July 2000; received in revised form 1 April 2001

Abstract

Fleet routing and flight scheduling are essential to a carrier's profitability, its level of service and its competitive capability in the market. This research develops a model and a solution algorithm to help carriers simultaneously solve for better fleet routes and appropriate timetables. The model is formulated as an integer multiple commodity network flow problem. An algorithm based on Lagrangian relaxation, a sub-gradient method, the network simplex method, the least cost flow augmenting algorithm and the flow decompositionalgorithm is developed to efficiently solve the problem. The results of a case study, regarding a major Taiwan airline's operations, show the model's good performance.

Scope and purpose

Fleet routing and flight scheduling issues have been widely studied to enhance airline's operation efficiency. Normally, network flow techniques are adopted for modeling and solving such complex mathematical problems. However, traditional approaches, which employ draft timetable as an essential medium, not only involve too much subjective judgement and decision making in the process but also reveal an incapability of directly and systematically managing the interrelation between supply and demand. The purpose of this paper is to develop a network model together with a solution algorithm, that can directly manage the interrelationships between passenger trip demands and flight supplies, in order to effectively assist carriers' scheduling. © 2002 Elsevier Science Ltd. All rights reserved.

Keywords: Time-space network; Integer multiple commodity network flow problem; Lagrangian relaxation; Least cost flow augmenting algorithm

*Corresponding author. Tel.: + 886-3-422-7151; fax: + 886-3-425-2960. E-mail address: t320002@cc.ncu.edu.tw (S. Yan).
1. Introduction

Fleet routing and flight scheduling are two critical activities in airline operations. In particular, they always affect aircraft usage efficiency, establishment of timetables, aircraft maintenance and crew scheduling. As a result, they are essential to a carrier’s profitability, its level of service and its competitive capability in the market. The flight scheduling process typically consists of two dependent phases: (a) the schedule construction phase and (b) the schedule evaluation phase. The construction phase is accomplished by drafting a timetable according to the projected demand, the market share, and the time slots of the available airports. After this, the draft timetable is then examined during the schedule evaluation phase for operating feasibility, cost and performance considerations. The feasibility checks in this evaluation phase mainly include the fleet routes, fleet size, crew scheduling, and maintenance arrangements. Any improvements identified during this phase are requested to be fed back into the construction phase to further revise the draft timetable. The flight scheduling process iterates between these two phases until a desirable timetable is obtained (Etshmaier and Mathaisel [1]).

Most airlines in Taiwan currently use a trial-and-error process for fleet routing and flight scheduling practices. They iterate the schedule construction and evaluation phases, as described above, manually. Such an approach is considered less efficient when the flight network becomes larger, and can possibly result in an inferior feasible solution.

Towards efficiency enhancement, Yan and Young [2] developed a decision support framework for multi-fleet routing and multi-stop flight scheduling. This framework was composed primarily of several strategic models which took account of a given draft timetable, a set of available airplanes, the airport slots, the airplane rental charges, and other cost data as basic input, so as to effectively solve for maximum profit.

Yan and Young’s approach [2], by developing mathematical models to effectively solve the two-phase iteration problem, was expected to be comparatively more systematic and efficient than the traditional trial-and-error method. Nevertheless, Yan and Young [2] still followed the above-mentioned two-phase-process, in which the very first draft timetable is originated from the planner’s personal experience. In such a procedure, the trip demand (OD) is firstly and roughly accommodated with the flight supply (i.e., airplane size, service frequency and route arrangement) manually. Afterwards, the strategic models are then applied to produce a better timetable within the resource constraints.

The draft timetable is an indispensable basis for current flight scheduling practices in Taiwan as well as in Yan and Young’s models [2], with the initial draft timetable being simply a rough estimation of the service frequency and routes for each OD pair. Such approaches, which employ draft timetable as an essential medium, not only involve too much subjective judgement and decision making in the process but also reveal an incapability of directly and systematically managing the interrelation between supply and demand.

In addition to Yan and Young’s study [2], there have been several types of airline scheduling models developed in these years, such as Abara’s [3] integer linear programming model for fleet assignment with fixed flight departure times, Teodorovic and Krcmar- Nozic’s [4] multicriteria model for deciding on flight frequencies under competitive conditions, the Balakrishnan et al. [5] mixed integer program model for long-haul routing, the Hane et al. [6] multicommodity network flow model for solving daily aircraft routing and scheduling problem (DARSP) without departure
time windows, and the Desaulniers et al. [7] set partitioning type model and time constrained multicommodity network flow model for solving DARSP problem according to a set of operational flight legs with known departure time windows. However, all of these models, as in Yan and Young’s [2], were basically developed with two-phase logic. As discussed above, such models, which make use of indirect approaches to handle supply-demand relationships, have a critical manual involvement in the process that is inclined to result in inaccuracy and inefficiency, which limits their use in practical operations.

The primary aim of this research then lies in developing both an integrated model and its solution algorithm, to help carriers with their multi-fleet routing and flight scheduling. To model and solve such problems more systematically, unlike the two-phase approach, neither draft timetable nor preset operational flight legs information is utilized in the proposed model. Instead, more accurate trip demands (OD) and all the supply restraints (e.g. aircraft types, fleet size and airport slots) serve directly as the model’s basic input. Thus, these inputs can easily be modified as required. Such a straightforward approach, as opposed to the two-phase-process, is expected to be more practical for carrier operations. How to develop a logical model as well as an efficient solution algorithm becomes the focus of this study.

The scope of the study is confined to the subject of pure fleet routing and flight scheduling operations under a given trip demand (OD), multiple aircraft types, fleet size, available slots, airport quotas and related cost data. Although the scheduling process, in practice, is closely related to the aircraft maintenance and the crew scheduling processes, these processes are usually separated to facilitate problem solving (Teodorovic [8]). Recently, Clarke et al. [9] tried to develop a fleet assignment model, which took maintenance and crew scheduling into considerations. This model was based on the Hane et al. [6] basic model handling only daily, domestic fleet assignment problems. However, these complicated maintenance and crew constraints are rather flexible in practice for the Taiwan airline studied in this research, due to its use of stand-by crews and its progressive maintenance policy (Yan and Young [2]). We therefore exclude these constraints in the modeling.

The fleet routing and flight scheduling model developed is formulated as an integer network flow problem with side constraints (NFPWS) which is characterized as an NP-complete problem (Garey and Johnson [10]). Although the branch-and-bound method and the cutting plane method are two typical solution techniques to exactly solve such problems (Levin [11]; Levin [12]; Teodorovic and Gubernic [13]; Lee [14]; Teodorovic [8]) these methods are apt to consume unpredicably long computation times even for a problem of normal size. Other techniques such as Dantzig–Wolfe decomposition (sometimes be referred as column generation method) and Lagrangian relaxation method were widely applied to solve such problems in these years (Ball et al. [15]). For example, Lee [14] tried a Lagrangian technique to solve a single fleet routing problem; however, the convergence results were far from satisfactory in Lee’s approach. The main reason for this was that finding good Lagrangian multipliers to improve the lower bound was difficult with the sub-gradient method (Fisher [16]). To solve the problem, Yan and Young [2] developed a modified sub-gradient method by integrating Fisher’s [16] and the Camerini et al. CFM methods [17]. This modified method led to a better performance, as observed from a comparison with Fisher’s and the CFM method in Yan and Young’s study [2]. Recently, there was much research adopted column generation methods for solving mixed integer programs (Barnhart et al. [18], Hane et al. [6], Desaulniers et al. [7]). Such technique may also be suitable for handling
the fleet assignment models in the likely studies, but one must be very careful of assuring the integer results.

The time–space network technique is very helpful in modeling the fleet routing and flight scheduling problems, as indicated by Yan and Young’s [2] studies. Hence, this popular technique is adopted for the same application in this research. In summary, this study applies a time–space network technique to formulate an integrated model for fleet routing and flight scheduling problems. An algorithm based on Lagrangian relaxation, Yan and Young’s [2] sub-gradient method, the network simplex method, the least cost flow augmenting algorithm and the flow decomposition algorithm is developed for solving the model. In particular, developing a good heuristic for finding a feasible upper bound (for minimization problems) and determining a proper sub-gradient method for modifying the Lagrangian multipliers to provide a good lower bound are two significant matters that directly affect the solution algorithm’s performance.

The following three sections will discuss, first, the modeling approach using a time–space network technique to formulate the model as an integer multiple commodity network flow problem. Then, a solution algorithm is developed to solve the proposed model. Finally, a case study is conducted to evaluate the performance of the model and the solution algorithm.

2. Modeling approach

A time–space network technique was applied to construct an integrated fleet routing and flight scheduling model for the purpose of maximizing a carrier’s profit. This model demanded the optimal management of both aircraft and passenger movements in the network through its systematic manipulation of direct flights, multi-stop flights and multiple fleets. The major elements in the modeling, including the fleet-flow time–space networks, the passenger-flow time–space networks, and the mathematical formulation, are as follows.

2.1. The fleet-flow time–space network

The study adopts several time–space networks to formulate the multi-fleet routing and flight scheduling problem. Each network indicates one specific type of aircraft’s potential movements within certain time periods and space locations, as shown in Fig. 1. The horizontal axis represents the airport locations; the vertical axis stands for time duration. “Nodes” and “arcs” are the two major components in the network. A node designates an airport at a specific time, while an arc represents an activity, such as a flight leg, a ground-holding, or an overnight stay. The arc flows express the flow of airplanes in the networks. Three types of arcs are defined below.

2.1.1. Flight leg arc

A flight leg arc represents a flight connecting between two different airports. All possible flight legs are installed into the network within a reasonable block of time, as long as time slots at the corresponding airports are available. Each flight leg arc contains information about the departure
time, the departure airport, the arrival time, the arrival airport, and the operating cost. The time block for a flight leg is calculated from the time when the airplane is prepared for this flight leg to the time when this flight leg is finished. Basically, it includes the time for investigation before departure, fuelling, passenger/baggage boarding and getting off, and flight time in the air. The flight cost is the arc cost in the network. The arc flow’s upper bound is one, meaning that the flight leg can be served at most once. The arc flow’s lower bound is zero, implying that no airplane serves this flight leg. In addition, a departure interval at the same airport is designed to be adjustable so as to meet the carrier’s operating requirements.
2.1.2. Ground arc

A ground arc represents the holding or the overnight stay of airplanes at an airport in a time window. The arc cost, including the airport tax, airport holding (or overnight stay) charge, gate use charge and other related cost, denotes the expenses incurred for holding an airplane at an airport in the corresponding time window. The arc flow’s upper bound is the apron capacity (or infinity, if the capacity is large), indicating the maximum number of airplanes that can be held at this airport during a specific time window. The arc flow’s lower bound is zero, implying that no airplane is held at this airport in this time window.

2.1.3. Cycle arc

A cycle arc functions to show the continuity between two consecutive planning periods. It connects the end of one period to the beginning of the next period for each airport. The arc flow’s upper bound and lower bound as well as the arc cost are the same as those of the ground arcs.

2.2. The passenger-flow time-space network

The time-space network technique is also applied to indicate passenger movements corresponding to certain times and locations, as shown in Fig. 2. Each passenger-flow time-space network represents a specific OD pair from the origin-destination table (known as the OD table). Such networks are designed to be symmetrical to the fleet-flow time-space networks so as to facilitate problem solving. The horizontal and vertical axes are defined as the same as those in the fleet-flow time-space network. A node, here, also represents an airport at a specific time. However, an arc designates an activity showing the passenger movement. Altogether, there are three types of arcs defined below.

2.2.1. Delivery arc

A delivery arc represents the passengers being transported from one airport to another on a flight. The transportation time is the same as the corresponding time block for the associated flight in the fleet-flow time-space network. The arc cost is the variable cost for serving each passenger (e.g. the catering cost). The arc flow’s upper bound is the capacity of the airplane (particularly the largest type), meaning that the maximum flow in the arc is the loading capacity of the aircraft. The arc flow’s lower bound is zero, indicating that no passenger from the corresponding OD is delivered on the associated flight.

2.2.2. Holding arc

A holding arc shows that the passengers stay at an airport in a time window. A waiting (or penalty) cost is the arc cost for the time window. However, if the arc just happens to connect either the departure or the arrival station of this network’s corresponding OD pair, the arc cost is then zero, because no passenger would, practically, make an unnecessary stay at the departure or arrival airports. Nevertheless, in practice the arc cost is adjustable. The arc flow’s upper bound is the station’s passenger service capacity within the network’s minimum time interval, implying that the maximum number of passengers can be accommodated at this airport in the time window. The arc flow’s lower bound is zero, showing that no passenger from the corresponding OD stays in the airport during the time window.
2.2.3. Demand arc

A demand arc connects the arrival station to the departure station of this network’s corresponding OD pair. It denotes the service demands for the OD pair that would actually be served in the network. The arc cost is the negative value of the average ticket fare. The arc flow’s upper bound is the projected demand for this OD pair. Aimed at maximizing a carrier’s profit, the passengers for this OD pair will not necessarily all be served in the model. The arc flow’s lower bound is zero, meaning that none of the OD pair’s passengers are served in the network. The trip demand for one specific OD pair could be flexibly divided into several demand arcs according to the actual demand.
distribution, the market characteristics, or the carrier’s own considerations. For example, the arcs could be designed to be denser for an OD pair, containing more commuter trips in the peak hours. In contrast, the arcs could be sparsely installed into the network if the passengers are less sensitive to time, such as for leisure trips. Time intervals for the demand arcs are adjustable. If the model’s results are expected to affect the original demand, one can change the inputs and rerun the model, until satisfactory results are acquired.

2.3. Notations of symbols used in the model formulation

Before introducing the model formulation, we first list the notations of symbols that will be used in the model formulation as follows:

- \( m \): the \( m \)th fleet;
- \( M \): the set of all fleets;
- \( n \): the \( n \)th OD pair;
- \( N \): the set of all ODs;
- \( Am \): the set of all arcs in the \( m \)th fleet network;
- \( Bn \): the set of all arcs in the \( n \)th passenger network;
- \( X_{ij}^m \): arc \((i, j)\) flow in the \( m \)th fleet network;
- \( Y_{ij}^n \): arc \((i, j)\) flow in the \( n \)th passenger network;
- \( C_{ij}^m \): arc \((i, j)\) cost in the \( m \)th fleet network;
- \( T_{ij}^n \): arc \((i, j)\) cost in the \( n \)th passenger network;
- \( NFm \): the set of all nodes in the \( m \)th fleet network;
- \( NPn \): the set of all nodes in the \( n \)th passenger network;
- \( CFm \): the set of all cycle arcs in the \( m \)th fleet network;
- \( AFm \): the number of available airplanes in the \( m \)th fleet network;
- \( FF \): the set of all flight arcs (integrating all fleet types);
- \( S^a \): the set of flight arcs at the \( a \)th station;
- \( Q^a \): the approved flight quota at the \( a \)th station;
- \( SA \): the set of all stations;
- \( K^m \): the aircraft capacity of the \( m \)th fleet network (note that a planning load factor could be used in the planning stage);
- \( U_{ij}^m \): arc \((i, j)\) flow’s upper bound in the \( m \)th fleet network;
- \( U_{ij}^n \): arc \((i, j)\) flow’s upper bound in the \( n \)th passenger network.

2.4. The model formulation

Based on the fleet-flow and the passenger-flow time space networks introduced above, we formulated the model as an integer network flow problem. There are several issues that need to be carefully considered: (1) the number of required airplanes in the network should not exceed the number of available airplanes for each fleet, (2) each flight can be served at most once in the fleet-flow networks, (3) the accumulation of flights for a certain period at one specific airport should not exceed its available quota, and (4) the number of transport passengers in a flight should never exceed the serving airplane’s capacity. Therefore, four types of side constraints are correspondingly designed during problem formulation: (1) the sum of the cycle arc flows in each fleet-flow network

...
shall not be greater than the number of available airplanes, (2) the sum of all the arc flows corresponding to the same flight should be equal to either one or zero, (3) the sum of the flights (including inflows and outflows from different fleet-flow networks) at each airport should not exceed its approved flight quota, and (4) the sum of all delivery arc flows corresponding to the same flight should not exceed the sum of each flight arc flows multiplied by the airplane capacity. The objective of this model is to “flow” the airplanes and passengers simultaneously in all networks at a minimum cost. Since the ticket revenue from the passenger-flow networks is in the form of a negative cost, this objective is equivalent to the maximization of profit. The model is formulated as follows:

Minimize

\[ Z = \sum_{m \in M} \sum_{ij \in Am} C_{ij}^m X_{ij}^m + \sum_{n \in N} \sum_{ij \in Bn} T_{ij}^n Y_{ij}^n \]

Subject to

(1) \[ \sum_{j \in NFm} X_{ij}^m - \sum_{k \in NFm} X_{ki}^m = 0, \quad \forall i \in NFm, \forall m \in M \]

(2) \[ \sum_{j \in NPn} Y_{ij}^n - \sum_{k \in NPn} Y_{ki}^n = 0, \quad \forall i \in NPn, \forall n \in N \]

(3) \[ \sum_{i \in CFm} X_{ij}^m \leq AFm, \quad \forall m \in M \]

(4) \[ \sum_{m \in M} X_{ij}^m \leq 1, \quad \forall ij \in FF \]

(5) \[ \sum_{m \in M} \sum_{i \in CF} X_{ij}^m \leq Q^n, \quad \forall a \in SA \]

(6) \[ \sum_{n \in N} Y_{ij}^n \leq \sum_{m \in M} K_{ij}^m X_{ij}^m, \quad \forall ij \in FF \]

(7) \[ 0 \leq X_{ij}^m \leq U_{ij}^m, \quad \forall ij \in Am, \forall m \in M \]

(8) \[ 0 \leq Y_{ij}^n \leq U_{ij}^n, \quad \forall ij \in Bn, \forall n \in N \]

(9) \[ X_{ij}^m \in \text{Integer}, \quad \forall ij \in Am, \forall m \in M \]

(10) \[ Y_{ij}^n \in \text{Integer}, \quad \forall ij \in Bn, \forall n \in N. \quad \text{(A)} \]

Model (A) is formulated as an integer multiple commodity network flow problem, in which the objective is to minimize the system cost. Constraints (1) and (2) ensure flow conservation at every
node in each fleet/passenger network, Eq. (3) denotes that the number of airplanes used in each fleet network should not exceed the available number of airplanes, Eq. (4) indicates that each flight is served at most once, Eq. (5) ensures that the sum of all flights at each station does not exceed its approved quota, Eq. (6) keeps the passenger delivery volume within the aircraft’s carrying capacity, Eqs. (7) and (8) hold all the arc flows within their bounds, and Eqs. (9) and (10) ensure the integrality of the airplane/passenger flows.

3. Solution algorithm

This research adopts Lagrangian relaxation along with a sub-gradient method to solve the proposed integer multiple commodity network flow problem. The solution process firstly relaxes the side constraints (Eqs. (3)–(6)) to construct the Lagrangian problem, and then solves it, to produce the optimal solution’s lower bound. Secondly, a heuristic, developed during this research, is applied to solve for the upper bound of the optimal solution. Then, a specific sub-gradient method, for revising the Lagrangian multipliers, is utilized to iterate the lower and upper bounds, until the convergence result is acceptable, or until the iteration count exceeds the preset number. These major parts of the solution algorithm (including the lower bound, the upper bound, and the sub-gradient method) are elaborated as follows:

3.1. The lower bound

The solution process for lower bound searching is shown in Fig. 4, and the steps are listed below. Step 1: The side constraints (3–6) are relaxed with the corresponding non-negative Lagrangian multipliers sets \( \mu_3, \mu_4, \mu_5, \) and \( \mu_6, \) and are added to the objective function of formula (A), resulting in the Lagrangian problem, (B). The optimal objective value for formula (B) becomes the lower bound of formula (A).

Minimize

\[
Z = \sum_{m \in M} \sum_{i \in \text{NFm}} C_{ij}^m X_{ij}^m + \sum_{n \in N} \sum_{i \in \text{Bn}} T_{ij}^n Y_{ij}^n + \sum_{m \in M} \left( \mu_3 \left( \sum_{i \in \text{CFm}} X_{ij}^m - AF_m \right) + \sum_{i \in \text{FF}} \left( \mu_4 \left( \sum_{m \in M} X_{ij}^m - 1 \right) \right) + \sum_{a \in S_A} \left( \mu_5 \left( \sum_{m \in M} \sum_{i \in \text{S'}} X_{ij}^m - Q^a \right) \right) \sum_{i \in \text{FF}} \left( \mu_6 \left( \sum_{n \in N} Y_{ij}^n - \sum_{m \in M} K^m X_{ij}^m \right) \right) \right)
\]

Subject to

(1) \( \sum_{j \in \text{NFm}} X_{ij}^m - \sum_{k \in \text{NFm}} X_{ki}^m = 0, \quad \forall i \in \text{NFm}, \forall m \in M \)

(2) \( \sum_{j \in \text{NPn}} Y_{ij}^n - \sum_{k \in \text{NPn}} Y_{ki}^n = 0, \quad \forall i \in \text{NPn}, \forall n \in N \)
\( (7) \quad 0 \leq X_{ij}^m \leq U_{ij}^m, \quad \forall ij \in Am, \forall m \in M \)

\( (8) \quad 0 \leq Y_{ij}^n \leq U_{ij}^n, \quad \forall ij \in Bn, \forall n \in N \)

\( (9) \quad X_{ij}^m \in \text{Integer}, \quad \forall ij \in Am, \forall m \in M \)

\( (10) \quad Y_{ij}^n \in \text{Integer}, \quad \forall ij \in Bn, \forall n \in N \) (B)

Step 2: Decompose formula (B) into two independent groups of networks, the fleet-flow networks and the passenger-flow networks.

Step 3: The fleet-flow networks are pure network flow problems and could also be characterized as minimum cost network flow problems. The network simplex method (NSM) is applied to efficiently solve these problems (see Kennington and Helgason [19] or Ahuja et al. [20]).

Step 4: The passenger-flow networks are also minimum cost network flow problems. However, since these problems involve only a single supply and demand in each network, the successive least cost flow augmenting path algorithm (SLCFAPAPA) may be employed to efficiently solve the problems. Each OD pair is loaded into each independent network using this method, and each network’s cost is then calculated.

Step 5: The lower bound for this iteration is finally obtained by aggregating all the fleet-flow and passenger-flow network costs.

3.2. The upper bound

A heuristic is developed to find an upper bound (a feasible solution). The searching process is indicated in Fig. 3, and the steps are listed below.

Step 1: Construct a new passenger-flow network (NPFN) in which each delivery arc flow is the sum of the corresponding delivery arc flows for all passenger-flow networks (layers) obtained in the lower bound searching process.

Step 2: Identify those delivery arcs in the NPFN that violate an aircraft’s carrying capacity. Then, repeatedly augment the extra volume of the violated arcs, layer by layer, by using the least cost flow augmenting path method with a modified label correcting algorithm (Powell [21]) until constraint (6) in formula (A) is satisfied.

Step 3: In each fleet-flow network, add the delivery arc’s ticket revenue in the NPFN to the flight cost so as to figure out the potential net profit (PNP) from each flight arc served by the corresponding type of aircraft.

Step 4: Establish a scheduling sub-problem by assigning the PNP to be the flight cost of fleet-flow networks. The modeling of the sub-problem refers back to Yan and Young’s research [2]. The objective is to minimize the system cost (min \( \sum CX \), fleet-flow networks only). The side constraints are the same as constraints (3–5) in formula (A), which are fleet size limitations, being served at most once for each flight, and the airport landing quota. The solution algorithm for this sub-problem also refers back to Yan and Young’s study [2]. The Lagrangian relaxation method, the network simplex method, the sub-gradient method and Yan and Young’s Lagrangian heuristic are all applied. As well, the
iteration limit, the acceptable convergence gap, and the initial value of the Lagrangian multipliers, are preset at this step.

Step 5: Apply the preset Lagrangian multipliers during this iteration to relax the side constraints of the sub-problem; then, utilize the network simplex method to solve for the lower bound solution of the sub-problem.
Step 6: Manipulate the following constraint, serving each flight at most once (constraint (4) in formula (A)). Sort the aircraft types according to their economy of scale, starting from the primary aircraft type network. First, identify those arcs that do not create the biggest PNP by comparing them to the same flight in other networks, and then set the arc cost of these flight arcs to be infinity. Secondly, solve this primary network again with the updated arc costs using the network simplex method to obtain the new network flows. Other aircraft network types would not only be handled with the same technique, but also be required to set their flight arc costs at infinity if the corresponding flight arcs in the previous networks are not zero.

Step 7: Manipulate the following constraint, the available landing quota for each airport (constraint (5) in formula (A)). Sum up each airport’s flights (departures and arrivals), and confirm any violation of the landing quota. Identify those arc flows to be reduced in the mth network (where m ≤ aircraft types), considering the cost, and set these arcs cost to be infinity. Also, set arc costs to be infinity in the mth network, if the flow is zero. Then again, apply the network simplex method to solve the mth network’s flows.

Step 8: Manipulate the fleet size constraint (constraint (3) in formula (A)). Sum up all cycle arc flows in the mth network, then check whether the number of aircraft used in the mth network has exceeded the mth fleet size (i.e. the “AFm” in formula (A)). Identify those cycle arcs where it is necessary to decrease their flows by considering airports with relatively more flights, and modify these cycle arcs’ upper bounds to be the required limits. Then, re-apply the network simplex method to finalize the mth network’s flows.

Step 9: Iterate steps 7 and 8 to ensure that all networks are examined and the airport landing quota, as well as fleet size constraints are fully satisfied.

Step 10: Compare the fleet-flow networks obtained in step 9 with the passenger-flow networks obtained in step 2, and subtract the passenger flows that have no corresponding flights to serve them.

Step 11: Measure the loading status of aircraft for all flights. Fill the surplus number of available seats by applying the least cost flow augmenting path algorithm with the corresponding OD that has not yet been served. Make sure that these loading increments do not violate the aircraft’s carrying capacity constraint (i.e. constraint (6) in formula (A)). The upper bound solution for this sub-problem is then obtained by aggregating the system costs from both fleet-flow and passenger-flow networks with the application of real arc costs into all networks.

Step 12: If the difference between the upper and the lower bound values of the sub-problem in this iteration does not converge to an acceptable gap, and the iteration counts have not yet exceeded the preset limit; then, update the Lagrangian multipliers using the sub-gradient method, and return to step 5.

The iteration process will not stop until the convergence result reaches a satisfactory degree, or the iteration counts have exceeded a preset number. Here, the upper bound solution acquired in the sub-problem is not only a feasible solution, but also an upper bound solution for the original problem.
3.3. The convergence mechanism

A modified sub-gradient method (Yan and Young [2]), for adjusting Lagrangian multipliers, is applied in this research, so as to obtain good convergence in the iteration results. The application steps are shown below.

Step 1: Set $c = 0$ and $\mu_0 = 0$.

Step 2: Solve the Lagrangian problem, (B), optimally by using the network simplex method and the successive least cost flow augmenting path algorithm so as to get a lower bound, $Z^L(\mu C)$. If the solution is feasible and also satisfies the complementary slackness condition, then this solution is an optimal solution; and, the solution process stops at this step. Otherwise, update the lower bound, $Z^L$.

Step 3: Apply the heuristic developed in this research to find an upper bound, $Z^U(\mu C)$, and update the upper bound, $Z^U$.

Step 4: If the gap between the lower bound ($Z^L$) and the upper bound ($Z^U$) falls within a specified tolerance ($\theta$), that is, $|Z^U - Z^L|/Z^U < \theta$, or the number of iterations reaches a limit, stop the algorithm.

Step 5: Adjust $\mu C$ by applying the modified sub-gradient method developed in Yan and Young’s study [2], to help improve the convergence. This modified method performs well, as indicated in Section 3.

Step 6: Set $c = c + 1$. Go to Step 2.

---

Table 1
Test results using exact solution algorithm

<table>
<thead>
<tr>
<th>Networks scale</th>
<th>Case A</th>
<th>Case B</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fleet-flow networks</td>
<td>3 layers</td>
<td>2 layers</td>
</tr>
<tr>
<td>Passenger-flow networks</td>
<td>6 layers</td>
<td>12 layers</td>
</tr>
<tr>
<td>Number of stations</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>Number of nodes</td>
<td>648</td>
<td>1344</td>
</tr>
<tr>
<td>Number of arcs</td>
<td>1329</td>
<td>3372</td>
</tr>
<tr>
<td>Models scale</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Variable number</td>
<td>1329</td>
<td>3372</td>
</tr>
<tr>
<td>1. integer</td>
<td>225</td>
<td>288</td>
</tr>
<tr>
<td>2. real</td>
<td>1104</td>
<td>3048</td>
</tr>
<tr>
<td>Flow conservation constraints</td>
<td>648</td>
<td>1344</td>
</tr>
<tr>
<td>Side constraints</td>
<td>153</td>
<td>294</td>
</tr>
<tr>
<td>Results</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PC type</td>
<td>Pentium 200</td>
<td>Pentium 300</td>
</tr>
<tr>
<td>CPU time</td>
<td>8 h 3 min 37 s</td>
<td>160 h 41 min 30 s</td>
</tr>
<tr>
<td>Number of iterations</td>
<td>4,073,794</td>
<td>106,978,684</td>
</tr>
</tbody>
</table>
| IP* found?                     | yes ($-45,978,000$)
| Integrality gap                | 0.58%           |                 |

---

*aThese tests were accomplished by using the LINGO software.

*bThe optimal objective value.
3.4. The miscellaneous

Because the solutions derived from the above process do not determine the route of each airplane, a flow decomposition algorithm (Ahuja et al. [20]) is accordingly recommended to decompose the link flows into arc chains. Each chain denotes an airplane’s rotation. Note that towards further reducing the convergence gap, a better lower bound solution than the aforementioned one may be solved linearly by using the LINGO software. That is to say, the original problem is transformed from an integer program (IP) to a linear program (LP) by relaxing the integer constraints (i.e. Constraints (9) and (10) in formula (A)). The optimal solution of this linear program (LP*) will theoretically function as a better lower bound solution for the original problem than what we can find from the Lagrangian problems (Yan [22]).

Furthermore, to evaluate the feasibility of applying exact solution algorithm to such a model, two small examples with different sizes were tested. Note that these networks were simplified by relaxing the integer constraints of all variables in passenger flow networks and the ground arc variables in fleet flow networks so as to make the problem to be more easily solved, and yet keep the models still satisfying planner’s needs. Through such simplifications, the original integer program (IP) was transformed into a mixed integer program (MIP). Two cases were tested as illustrated in Table 1. The computational results show that it is difficult to exactly solve even a small size problem of such a model structure by using a commercial optimization solver.

4. Case study

To demonstrate and to test how well the model and the solution algorithm may be applied to the real world, we perform a case study regarding a major Taiwan airline’s operations.

4.1. The input

Our case study was based on data obtained from a major Taiwan airline’s domestic operations during 1996 (TransAsia Airways). There were 11 cities served by about 170 daily flights during its domestic operations. Three types of aircraft were used, including AirBus 320s, AirBus 321s, and ATR 72s. For simplification, all the aircraft were categorized into two fleets: fleet A included all the AirBus series’ aircraft (12 airplanes), with an average of 178 seats each, and fleet B contained all the ATR 72 aircraft (12 airplanes) with 72 seats each. All the cost parameters in the fleet-flow and the passenger-flow time–space networks were set according to airline reports and the Taiwan government regulations, with reasonable simplifications. Moreover, iterations during the solution process were limited to be less than 1000 times. Note that the cost used here is the direct operating cost. The fixed cost (or the sunk cost) and the indirect cost (for example, aircraft’ capital investment, depreciation, maintenance or rental charges) are not included in the model. In other words, the profit calculated here is short-term “operating profit” rather than the actual profit of the system. However, through the optimization of the short-term profit, system profits can be enhanced over the long-term from system perspectives.

According to the carrier’s operation statistics for 1996, there were in total about 12,800 trips being served each day. Since the case study covered a domestic operation, the operating period was
set to be one day. As well, the planning load factor for each flight was set not to exceed 0.7. Other inputs, such as the flight time, the distance between stations, the landing quota, the available time slots, and the ground handling time, were mainly based on actual operation data, referred to related reports.

4.2. The output

This case study contained two layers of fleet-flow time–space networks as well as 34 layers of passenger-flow time–space networks, involving 9504 nodes and 25,558 arcs (variables). The model, which is substantially large in terms of combinatorial optimization, included 10,407 constraints in which 9504 constraints were for ensuring flow conservation and 903 side constraints.

The results indicated that over 99% of the trips (12,760 trips) were served, and that only 12 fleet A airplanes as well as 5 fleet B airplanes were required to optimally provide 108 flights each day. The planned flight number dropped sharply compared to flights offered during the actual operations, because the average load factor during actual operations was only about 0.5. Moreover, passenger flow results showed that 94% of the passengers were transported by fleet A, but only 6% of the passengers were carried by fleet B. This indicated that fleet A, in general, was more profitable than fleet B. Moreover, all passengers completed their trips on non-stop flights, i.e. no transfer movement occurred in the system. This was probably due to the small network scale and the short line-haul distances in the case study. The OD distances were all within 330 km. It would seem reasonable, from the economic point of view, to have only direct flights for handling such flight lengths. Although there were no transfer passengers (i.e. no multi-stop flights) in this case study, the effective assignment of passengers and fleets showed the capability of this integrated model to enhance carrier operations.

The model converged to within a 2.5% error gap, as shown in Fig. 4, in 5287 s of CPU (Pentium 200) time. The best upper bound, the objective value, was NT$-8,724,820 (obtained at iteration 548), while the best lower bound was NT$-8,949,754 (obtained at iteration 199). Moreover, the problem was solved linearly (by LINGO) so as to have a better lower bound solution (LP*), NT$-8,936,300 (in 2546 s of CPU time). Thus, the error gap could be further converged to within 2.4%. These results also indicate that the lower bound solution obtained in this research, by applying the Lagrangian relaxation algorithm with the sub-gradient method, was already very close to the linear-relaxed lower bound solution, i.e. the linear optimal solution (LP*).

The fleet flows obtained above cannot yet be directly put to practical use without identifying each airplane’s route in the fleet networks. The flow decomposition method (Ahuja et al. [20]) was then applied to trace the path of each airplane. The fleet flows in the networks were decomposed into several arc chains, each representing an airplane’s rotation, as shown in Fig. 5.

4.3. Sensitivity analyses

Because the algorithm developed in this research was a problem-oriented heuristic, its performance might possibly be affected by problem parameters. To confirm its performance, and to understand the influence of the parameters on the solution, sensitivity analyses of such critical inputs as the fleet size, the demand (OD), the flight cost and the ticket price were conducted.
4.3.1. Fleet size

Case study results indicated that the carrier by providing 12 airplanes from fleet A and 5 airplanes from fleet B, would lead to a maximum profit in short term operations. For this profit, more than 99% of the demand (OD) could be served; that is, the service rate (SR) would be larger than 99%. As shown in Fig. 6, an increase in fleet size, whether for fleet A or fleet B, did not improve the objective value or the service rate. It should be noted that in this research the flight cost considered only the variable cost directly related to flight operations, meaning that extra aircraft incurred no extra cost. However, decreasing aircraft from fleet A or fleet B (from 12A + 5B) would make the profit fall off 3.2% and 0.4%, respectively. Obviously, a decrease in the size of fleet A was much more sensitive than for fleet B, meaning that type A aircraft are more valuable than type B aircraft. In addition, the service rate (SR) dropped sharply from 98% to 87% when the size of fleet A was decreased from 11 airplanes to 8 airplanes.

4.3.2. Demand

In practice, trip demands (OD) would change along with the future market. As shown in Fig. 7, when the demand increased by less than 20%, the supply, 12 airplanes from fleet A and 5 airplanes from fleet B, could sustain a 98–99% service rate. Every percentage of demand increment would add NT$71,000–92,000 to the objective value. On the other hand, when the demand decreased by less than 20%, then 11 airplanes from fleet A and 5 airplanes from fleet B would be enough to serve 99% of the total trips. Every percentage of demand decline would correspondingly reduce NT$93,000 of the objective value. In general, an analysis of demand fluctuation between ±20% would result in considerable variations in the objective value. That is, a 1% decrease in demand would cut approximately 1% from the objective value.

4.3.3. Flight cost

The flight cost was of critical input to the model proposed in this research. When the flight cost oscillated between ±20%, as shown in Fig. 8, it led to a smooth change in the objective value. An
approximate 1% increase in the flight cost would cut 0.3% from the objective value which was about NT$25,000 on average. It seemed then that the flight cost would maintain an almost linear relationship with the objective value.
4.3.4. **Ticket price**

The ticket price directly affected carriers’ revenue. The price ceiling was regulated but in actual operations, carriers were authorized to offer price discounts to the market. The input ticket price was set to be about 70% of the list price (the ceiling) in the basic model. Here, a ± 20% difference in the ticket price was tested to examine the sensitivity. As shown in Fig. 9, an increase in ticket price of less than 20% would still maintain a 99% service rate, while a 20% decrease in ticket price would result in a 100% service rate. The resulting service rates were similar for all scenarios, possibly because the ticket price was relatively high compared to the operating cost. The tests also indicated that a 1% ticket price variation would make a 1.3% change in the objective value, which was, on average, equal to NT$117,000.

4.4. **Applications**

The model and the solution algorithm developed in this research were anticipated to function as an effective planning tool for fleet routing and flight scheduling. This tool could be applied to
simulate changes in key factors, which could be mainly controlled by the carriers, such as the fleet size, the flight cost and the catering cost. Also, it could be employed to analyze a system's performance when some critical environmental elements are expected to change, for example the future demands, flexible pricing for market competition, available airport take-off/landing quotas.
and available time slots. In general, the application of the model should be very comprehensive. Table 2 shows one application example. Carriers could assemble all sorts of airport combinations and check each combination’s profitability as well as the corresponding resources required. Such types of analysis should present a more integrated and quantitative evaluation, and the results should be helpful for carriers competing for new flight rights.

There were in total 11 Taiwan domestic airports covered in this example as shown in Table 2. They were Taipei, Kaohsiung, Hwalien, Maku, Tainan, Chiyi, Kinmen, Pintung, Taitung, CKS, and Hsinchu. The problem sizes tested are up to 25,558 variables and 10,407 constraints. The error gaps are within 2.4%, showing the good performance of our solution algorithm. In these eleven scenarios, S3, as shown in Table 2, represented an alternative containing the first three airports, i.e. Taipei, Kaohsiung, and Hwalien. S4 denotes an alternative involving the first four airports, namely Taipei, Kaohsiung, Hwalien, and Maku. The rest were the same. For example, if the carrier is situated in scenario “S8”, and there is a bid for flight rights from/to Pintung (the 9th airport), then, “S9” in Table 1 would reveal that the carrier does not need to increase its fleet size (11A + 3B) for those extra flights from/to Pintung, and there would be an approximate 3% increase in revenue (\(8272 - 8021)/8021\)). Note that other analyses can be similarly performed. Such information, derived from the model’s application, should be very helpful for the carriers in their decision making process.

5. Conclusions

The research that has contributed most to fleet routing and flight scheduling is the development of a model capable of directly managing the interrelationship between trip demands and flight supplies. This model differs from the conventional approach, i.e., using a draft timetable as an indispensable medium, conducting the scheduling process through the schedule construction and the evaluation phases. The modeling approach adopted in this research contains theoretically less manual involvement, which has inclined towards inaccuracy and inefficiency in actual operations.

The second contribution of the study is the development of an efficient solution algorithm for the proposed model. The model is formulated as an integer multiple commodity network flow problem. The model’s solution algorithm is mainly based on the application of the Lagrangian relaxation method, the network simplex method, a modified sub-gradient method, the least cost flow augmenting path algorithm, and a self-developed upper bound heuristic. In addition, a flow decomposition algorithm is applied to trace each aircraft’s route.

A case study utilizing the domestic operations of a major Taiwan airline was conducted to show how to actually apply the model in the real world. The case study contained two different fleets with a total of 24 aircraft (12A + 12B) serving approximately 12,800 daily trips, among 11 airports. The network included 9504 nodes and 25,558 arcs, which made a problem size of up to 25,558 variables, with 10,407 constraints. The solution algorithm performed well, being capable of converging within an error gap of 3%. The results indicate that the fleet size could be reduced to 12A + 5B to maximize carrier’s profit. Also, in the case study there were several critical sensitivity tests, as well as a specific application being carried out, to demonstrate the flexible use of the model.

Although the preliminary test results show that the model and the solution algorithm are potentially useful for scheduling operations, especially for Taiwan domestic airlines, more tests or
case studies should be conducted so that carriers may grasp its limitations before putting it to practical use. For larger airlines in other area, for example, American Airlines, the model may be applied with suitable modifications. The solution algorithm may as well be modified if the convergence is not satisfactory for solving large-scale problems. For example, the upper bound heuristic and/or the sub-gradient method may be improved. Modern meta-heuristic techniques, for example, tabu search method or threshold accepting method, may be incorporated into the algorithm to improve the algorithm. Furthermore, other useful algorithms for combinatorial optimization, for example, column generation, branch-and-price or genetic algorithm, may be developed to help solve large-scale problems. All of these could be directions of future research.

Finally, although the model and the solution algorithm developed are anticipated to function as a helpful tool for airline planning practices, the model's applications should not be restricted to only air transport. Through partial modifications, the model should be able to be applied to other problems or modes with similar characteristics, for example, fleet routing and shipping scheduling in marine transport, the schedule design of highway passenger transport, and the scheduling of an intermodal cargo transportation system. These potential applications could presumably form future research topics.

Acknowledgements

This research was supported by a grant (NCHC-86-08-008) from the National Science Council of Taiwan. We would like to thank TransAsia Airways for providing the test data and valuable opinions on this research. We also thank the two anonymous referees for their helpful comments and suggestions on the presentation of the paper.

References


Shangyao Yan is a professor and the chair in the Department of Civil Engineering, National Central University, Taiwan. He received both his Bachelor and Master degrees from National Taiwan University. He obtained his Ph.D. from Massachusetts Institute of Technology, USA. His current research topics are focused on airline/airport operations research, network design and combinatorial optimization.

Chih-Hwang Tseng received the Bachelor degree from Chun-Yuan University, the Master degree from National Chiao-Tung University, the Ph.D. degree from National Central University, Taiwan. He is working at the Institute of Transportation, Ministry of Transportation and Communications of Republic of China. His research interests are transportation planning, system optimization and project evaluation.