World Academy of Science, Engineering and Technology
International Journal of Mathematical and Computational Sciences
Vol:4, No:6, 2010

Applying Lagrangian Relaxation-Based Algorithm for the Airline Coordinated Flight Scheduling Problems

Chia-Hung Chen, Shangyao Yan

***Abstract*—**The solution algorithm, based on Lagrangian relaxation, a sub-gradient method and a heuristic to find the upper bound of the solution, is proposed to solve the coordinated fleet routing and flight scheduling problems. Numerical tests are performed to evaluate the proposed algorithm using real operating data from two Taiwan airlines. The test results indicate that the solution algorithm is a significant improvement over those obtained with CPLEX, consequently they could be useful for allied airlines to solve coordinated fleet routing and flight scheduling problems.

***Keywords*—**Coordinated flight scheduling, Multiple commodity network flow problem, Lagrangian relaxation

Open Science Index, Mathematical and Computational Sciences Vol:4, No:6, 2010 publications.waset.org/10886/pdf

I. Introduction

FLEET routing and flight scheduling are a fundamental issue for airlines, because they are essential to a carrier's profitability, its level of service and its competitiveness in the market. In practice, the flight scheduling process typically consists of two dependent phases: (a) the schedule construction phase; and (b) the schedule evaluation phase (Etshmaier and Mathaisel[1]). During the flight scheduling process these two phases are iterated until a desirable timetable is obtained. The process becomes less efficient when the flight network becomes larger, and can possibly result in an inferior feasible solution.

Studies on fleet routing and flight scheduling for passenger transportation have been performed by many, for example, by Yan and Tseng[2], Barnhart et al.[3], Lohatepanont and Barnhart[4], and Yan et al.[5]. Apart from fleet routing and flight scheduling for passenger transportation, there has also been significant research devoted to freight transportation and fleet routing, for examples, see Yan et al.[6]. Models have been formulated as integer linear programs, mixed integer programs or multiple commodity network flow problems. The above problems have usually been solved using exact solution methods, such as the simplex method, the branch-and-bound technique, the cutting plane method, the Lagrangian relaxation-based algorithm, and the column and row generation technique.

The main focus in the above studies was on single carrier transportation. However, in recent years, international strategic alliances have come to being in a broad spectrum of industries, especially in the airline industry. The setting of a good coordinated flight schedule can not only enhance the operating performance of the allied airlines, but can also act as a useful reference for decision-making. However, currently, most Taiwan airline alliances use a trial-and-error process for fleet routing and flight scheduling. They iteratively construct and evaluate the schedule phases manually and independently of each other, without optimization from a systemic perspective, after which it is necessary for the allied airlines to check whether these schedules are mutually acceptable. If not, each schedule is modified further. The process is repeated until satisfactory results are obtained. Such an approach is neither effective nor efficient, especially when the flight network becomes large. Inferior solutions can be the result.

Constraints arising from each allied airline's schedule make the scheduling problem more complicated. How to simultaneously determine a good coordinated flight schedule that satisfies each airline in the alliance is even more difficult. Yan and Chen[7, 8] employed network flow techniques to construct coordinated scheduling models for passenger- and cargo-transportation, respectively. These models are formulated as mixed integer multiple commodity network flow problems with side constraints (NFPWS) that are characterized as NP-hard (Garey and Johnson[9]). Problem sizes are expected to be huge making the model more difficult to solve than traditional passenger/cargo flight scheduling problems. For example, it took 5547.38 seconds to solve a small-scale problem for a mixed alliance with only eight stations and four airplanes for Airline r_1, and ten stations and six airplanes for Airline r_2 (Yan and Chen[8]). Clearly, more efficient solution algorithms for solving large-scale problems have to be developed.

The branch-and-bound method and the cutting plane method are two typical solution techniques that have been used in the past to exactly solve fleet routing and flight scheduling problems, for example, see Lee[10]; Teodorovic[11]. The branch-and-bound method consists of a systematic enumeration of all candidate solutions, where large subsets of fruitless candidates are discarded together, by using the upper and lower estimated bounds of the quantity being optimized. The cutting plane method works by solving the non-integer linear program, the relaxation of the given integer program, and then testing whether the optimum found is also an integer solution. If this is not the case, a new restriction is added that cuts off the non-integer solution but does not cut off any integer points of the feasible region. This is repeated until an optimal integer solution is found. In other words, the above two methods require effort that grows exponentially with problem size. Other techniques such as the Dantzig-Wolfe decomposition (sometimes referred to as the column generation method) and the Lagrangian relaxation methods have also been widely applied to solve such problems in recent years (Ball et al.[12]). For example, Lee[10] tried a

Authors are with Shu-Te University, Taiwan. e-mail: chiahung@stu.edu.tw

Open Science Index, Mathematical and Computational Sciences Vol:4, No:6, 2010 publications.waset.org/10886/pdf

Lagrangian technique to solve a single fleet routing problem; however, the convergence results were far from satisfactory. The main reason for this is that finding good Lagrangian multipliers to improve the lower bound is difficult with the sub-gradient method (Fisher[13]). To solve the problem, Yan and Young[14] developed a modified sub-gradient method by integrating Fisher[13]'s and Camerini et al.[15]'s CFM methods. This modified method led to a better performance, as observed from a comparison with Fisher's and the CFM method in Yan and Young[14]'s study. To improve Yan and Young[14]'s model, Yan and Tseng[2] developed an integrated scheduling model and a solution algorithm, mainly based on Lagrangian relaxation, the network simplex method, a modified sub-gradient method, the least cost flow augmenting path algorithm, and a self-developed upper bound heuristic, to help carriers simultaneously solve for better fleet routes and appropriate timetables. Recently, column generation and branch and bound methods have been commonly adopted for solving mixed integer programs (e.g., the branch and price method; see Barnhart et al.[16]).

Following recent efforts made to solve single carrier fleet routing and flight scheduling problems (e.g., Yan and Young[14] and Yan and Tseng[2]), we develop the Lagrangian based algorithm, which is based on a combination of Lagrangian relaxation, Yan and Tseng[2]'s sub-gradient method, and a heuristic for finding the upper bound solution, designed to solve coordinated fleet routing and flight scheduling problems.

The remainder of this paper is organized as follows: first, we describe the problem, then develop the solution algorithm for solving the problem. Thereafter, a case study is conducted to evaluate the performance of the solution algorithm. Finally, we give some conclusions.

II. Problem Description

Recently, Yan and Chen[7, 8] employed network flow techniques to construct coordinated scheduling models for passenger- and cargo-transportation, respectively. The more complicated coordinated cargo fleet routing and flight scheduling problem is adopted to demonstrate the performance of the Lagrangian based algorithms. Based on real practices, Yan and Chen[8] constructed the alliance-based networks that would occur under swap space and complementary alliances. Their aim was to develop coordinated scheduling models to set the most satisfactory cargo fleet routes and timetables for allied airlines. The two major elements in Yan and Chen's fleet-flow and the cargo-flow networks indicate the fleet and cargo movements within a specified time period (one week in Yan and Chen[8]) and space locations. The horizontal axis represents the airport locations; the vertical axis stands for the time duration. All available airports are included. "Nodes" and "arcs" are the two major components in the network. For a more detailed description of the problem, please see Yan and Chen[8]. Before formulating the model, let us define the following symbols and notations:

r : the r^{th} allied airline;

R : the set of all allied airlines. In this study, $R = \{r_1, r_2\}$;

n : the n^{th} OD pair;

N^r : the set of all ODs for the r^{th} allied airline;

m : the m^{th} complementary OD pair;

CCN : the set of all complementary alliance cargo-flow networks for all allied airlines;

A^r, NF^r : the set of all arcs and nodes in the r^{th} fleet-flow network;

CF^r : the set of all cycle arcs in the r^{th} fleet-flow network;

BF^{nr} : the set of all demand arcs in the n^{th} cargo-flow network of the r^{th} allied airline;

B^{nr}, NP^{nr} : the set of all arcs and nodes in the n^{th} cargo-flow network of the r^{th} allied airline;

AF^r : the number of available airplanes for the r^{th} allied airline;

FF^r : the set of all flight arcs in the r^{th} fleet-flow network;

IFF^r : the set of all flight arcs in the r^{th} fleet-flow network, associated with the flights serving only the r^{th} allied airline's cargos but not the other allied airline's cargos;

S_g^r : the set of the r^{th} allied airline flight arcs that arrive at (or departure from) the g^{th} station;

S_{gh}^r : the set of the r^{th} allied airline flight arcs that connect the g^{th} station to the h^{th} station;

Q_g^r : the approved flight quota (arrivals or departures) at the g^{th} station for the r^{th} allied airline;

Q_{gh}^r : the approved flight quota that connects the g^{th} station to the h^{th} station for the r^{th} allied airline;

K^r : the aircraft capacity of the r^{th} allied airline (in weight units for the studied airlines, a planning factor may be considered);

SA^r : the set of all stations for the r^{th} allied airline;

SAB^r : the set of airport pairs with an approved flight quota for the r^{th} allied airline;

FU_{ij}^r : the arc(i,j) flow's upper bound in the r^{th} fleet-flow network;

CU_{ij}^{nr} : the arc(i,j) flow's upper bound in the n^{th} cargo-flow network for the r^{th} allied airline;

$CCDA^{mr}$: the set of the demand arcs in the m^{th} complementary alliance cargo-flow network for the r^{th} allied airline;

C_{ij}^{r} : the arc(i,j) cost in the r^{th} fleet-flow network;

T_{ij}^{nr} : the arc(i,j) cost in the n^{th} cargo-flow network for the r^{th} allied airline;

F_i^r : a fixed cost of the r^{th} allied airline for choosing station i, either as a departure airport or an arrival airport;

V_i^r : a variable cost of the r^{th} allied airline at station i to handle cargos per unit weight, including loading and unloading;

D_{ij}^{m} : the projected demand associated with the demand arc(i,j) in both airlines' m^{th} complementary alliance cargo-flow networks;

M : a very large value;

X_{ij}^{r} : the arc(i,j) flow in the r^{th} fleet-flow network;

Y_{ij}^{nr} : the arc(i,j) flow in the n^{th} cargo-flow network for the r^{th} allied airline;

W_i^r : a decision variable of the r^{th} allied airline to choose station i for service, which equals 1 if station i is served, and 0 otherwise;

Based on the fleet-flow and the cargo-flow time space networks, as well as the side constraints, Yan and Chen[8] formulated the model as a mixed integer network flow problem. The objective of the model is to "flow" the airplanes and cargos simultaneously in all networks at a minimum cost. Since the revenue from the cargo-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:

Model (A):

Minimize

$$Z=\sum_{r}\sum_{ij\in A^r}\left(C_{ij}^{r}X_{ij}^{r}\right)+\sum_{r}\sum_{n\in N^r}\sum_{ij\in B^{rn}}\left(T_{ij}^{nr}Y_{ij}^{nr}\right)+\sum_{r}\sum_{i\in SA^r}\left(F_i^r W_i^r\right)+\sum_{r}\sum_{n\in N^r}\sum_{ij\in BF^{nr}}\left[Y_{ij}^{nr}\left(V_i^r+V_j^r\right)\right] \quad (1)$$

Subject to

$$\sum_{j\in NF^r}X_{ij}^{r}-\sum_{k\in NF^r}X_{ki}^{r}=0 \qquad \forall i\in NF^r,\forall r\in R \quad (2)$$

$$\sum_{j\in NP^{nr}}Y_{ij}^{nr}-\sum_{k\in NP^{nr}}Y_{ki}^{nr}=0 \qquad \forall i\in NP^{nr},\forall n\in N^r,\forall r\in R \quad (3)$$

$$\sum_{ij\in CF^r}X_{ij}^{r}\le AF^r \qquad \forall r\in R \quad (4)$$

$$\sum_{ij\in S_g^r}X_{ij}^{r}\le Q_g^r \qquad \forall g\in SA^r,\forall r\in R \quad (5)$$

$$\sum_{ij\in S_{gh}^r}X_{ij}^{r}\le Q_{gh}^r \qquad \forall gh\in SAB^r,\forall r\in R \quad (6)$$

$$\sum_{n\in N^r}Y_{ij}^{nr}\le K^r X_{ij}^{r} \qquad \forall ij\in IFF^r,\forall r\in R \quad (7)$$

$$\sum_{n\in N^{r_1}}Y_{ij}^{nr_1}+\sum_{n\in N^{r_2}}Y_{ij}^{nr_2}\le K^r X_{ij}^{r} \qquad \forall ij\in FF^r\setminus IFF^r,\forall r\in R \quad (8)$$

$$\sum_{r\in R}Y_{ij}^{mr}\le D_{ij}^{m} \qquad \forall ij\in\bigcup_{r}CCDA^{mr},\forall m\in CCN \quad (9)$$

$$\sum_{n\in N^r}\sum_{ij\in BF^{nr}}Y_{ij}^{nr}\le MW_i^r \qquad \forall i\in SA^r,\forall r\in R \quad (10)$$

$$0\le X_{ij}^{r}\le FU_{ij}^{r} \qquad \forall ij\in A^r,\forall r\in R \quad (11)$$

$$0\le Y_{ij}^{nr}\le CU_{ij}^{nr} \qquad \forall ij\in B^{nr},\forall n\in N^r,\forall r\in R \quad (12)$$

$$X_{ij}^{r}\in I \qquad \forall ij\in A^r,\forall r\in R \quad (13)$$

$$W_i^r\in 0\ or\ 1 \qquad \forall i\in SA^r,\forall r\in R \quad (14)$$

Open Science Index, Mathematical and Computational Sciences Vol:4, No:6, 2010 publications.waset.org/10886/pdf

Open Science Index, Mathematical and Computational Sciences Vol:4, No:6, 2010 publications.waset.org/10886/pdf

The model is formulated as a mixed integer multiple commodity network flow problem, in which the objective is to minimize the total system cost of the allied airlines. Constraints (2) and (3) ensure flow conservation at every node in each fleet/cargo-flow network. Constraint (4) denotes that the number of airplanes used in each fleet-flow network should not exceed the available number of airplanes. Constraint (5) ensures that the sum of all flights at each station does not exceed its approved quota. Constraint (6) ensures that the sum of all flight arcs connecting the gth station to the hth station does not exceed the approved flight quota. Constraint (7) keeps the cargo delivery volume within the aircraft's carrying capacity for flights carrying only an allied airline's cargo. Constraint (8) keeps the cargo delivery volume within the aircraft's carrying capacity for the flights serving both the airline's and its ally's cargos. Equation (9) indicates that the sum of the demand arc flows in the two networks, associated with the same demand, should be less than or equal to the projected demand. Equation (10) is used to determine whether a station, origin or destination, is used for serving cargos, or not. That is, if station i is used for serving the r^{th} allied airline's cargos, then $W_i^r = 1$; otherwise $W_i^r = 0$. Constraints (11) and (12) hold all the arc flows within their bounds. Equation (13) ensures the integrality of the airplane flows. Constraint (14) indicates that each airport selection decision is binary.

Model (B):

Minimize

$$
\begin{aligned}
Z = & \sum_r \sum_{ij \in A^r} C_{ij}^r X_{ij}^r + \sum_r \sum_{n \in N^r} \sum_{ij \in B^{nr}} T_{ij}^{nr} Y_{ij}^{nr} + \sum_r \sum_{i \in SA^r} F_i^r W_i^r + \sum_r \sum_{n \in N^r} \sum_{ij \in BF^{nr}} Y_{ij}^{nr} \left(V_i^r + V_j^r \right) \\
& + \sum_{r \in R} \left(\mu 4^r \left(\sum_{ij \in CF^r} X_{ij}^r - AF^r \right) \right) + \sum_{r \in R} \sum_{g \in SA^r} \left(\mu 5_g^r \left(\sum_{ij \in S_g^r} X_{ij}^r - Q_g^r \right) \right) \\
& + \sum_{r \in R} \sum_{gh \in SAB^r} \left(\mu 6_{gh}^r \left(\sum_{ij \in S_{gh}^r} X_{ij}^r - Q_{gh}^r \right) \right) + \sum_{r \in R} \sum_{ij \in IFF^r} \left(\mu 7_{ij}^r \left(\sum_{n \in N^r} Y_{ij}^{nr} - KX_{ij}^r \right) \right) \\
& + \sum_{r \in R} \sum_{ij \in FF^r \setminus IFF^r} \left(\mu 8_{ij}^r \left(\sum_{n \in N^{r_1}} Y_{ij}^{nr_1} + \sum_{n \in N^{r_2}} Y_{ij}^{nr_2} - KX_{ij}^r \right) \right) + \sum_{m \in CCN} \sum_{ij \in \bigcup_r CCDA^{mr}} \left(\mu 9_{ij}^m \left(\sum_{r \in R} Y_{ij}^{mr} - D_{ij}^m \right) \right)
\end{aligned}
\tag{15}
$$

subject to constraints (2), (3), (10), (11), (12), (13), and (14).

Step 2 : Model (B) is decomposed into four independent groups of networks, such as Model (C), the cargo-fleet-flow network for Airline r_1, Model (D), the cargo-fleet-flow network for Airline r_2, Model (E), the cargo-flow networks for Airline r_1, Model (F), and the cargo-flow networks for Airline r_2.

Model (C)

III. Development of the Solution Algorithm

The Lagrangian based algorithm is developed to efficiently solve coordinated fleet routing and flight scheduling problems. Referring to the concepts in Yan and Tseng[2], we adopt the Lagrangian relaxation technique, coupled with a sub-gradient method, to develop a Lagrangian based algorithm to solve the problem. The solution processes of the algorithms are the same. We first relax the side constraints (constraints (4) to (9)) to construct the Lagrangian problem, and then solve it, to procure the lower bound of the optimal objective. Secondly, we develop a heuristic to solve for the upper bound of the optimal objective. Then a sub-gradient method for revising the Lagrangian multipliers is utilized to iterate the lower and upper bounds, until an acceptable convergence result is reached, or until the computation time exceeds the pre-set time limit.

A. The lower bound solution

The steps for searching for the lower bound are listed below:

Step 1 : Side constraints (4) to (9) are relaxed with the corresponding non-negative Lagrangian multiplier sets, $\mu 4$ to $\mu 9$, and are added to the objective function of Model (A), resulting in Model (B). The optimal objective value for Model (B) becomes the lower bound of Model (A).

Minimize

$$Z = \sum_{ij \in A^{r_1}} C_{ij}^{r_1} X_{ij}^{r_1} + \mu 4^{r_1} \sum_{ij \in CF^{r_1}} X_{ij}^{r_1} + \sum_{g \in SA^{r_1}} \mu 5_g^{r_1} \sum_{ij \in S_g^{r_1}} X_{ij}^{r_1} + \sum_{gh \in SAB^{r_1}} \mu 6_{gh}^{r_1} \sum_{ij \in S_{gh}^{r_1}} X_{ij}^{r_1} - \sum_{ij \in IFF^{r_1}} \mu 7_{ij}^{r_1} KX_{ij}^{r_1} - \sum_{ij \in FF^{r_1} \setminus IFF^{r_1}} \mu 8_{ij}^{r_1} KX_{ij}^{r_1} - \mu 4^{r_1} AF^{r_1} - \sum_{g \in SA^{r_1}} \mu 5_g^{r_1} Q_g^{r_1} - \sum_{gh \in SAB^{r_1}} \mu 6_{gh}^{r_1} Q_{gh}^{r_1} \quad (16)$$

subject to constraints (2), (11), and (13).

Model (D)

Minimize

$$Z = \sum_{ij \in A^{r_2}} C_{ij}^{r_2} X_{ij}^{r_2} + \mu 4^{r_2} \sum_{ij \in CF^{r_2}} X_{ij}^{r_2} + \sum_{g \in SA^{r_2}} \mu 5_g^{r_2} \sum_{ij \in S_g^{r_2}} X_{ij}^{r_2} + \sum_{gh \in SAB^{r_2}} \mu 6_{gh}^{r_2} \sum_{ij \in S_{gh}^{r_2}} X_{ij}^{r_2} - \sum_{ij \in IFF^{r_2}} \mu 7_{ij}^{r_2} KX_{ij}^{r_2} - \sum_{ij \in FF^{r_2} \setminus IFF^{r_2}} \mu 8_{ij}^{r_2} KX_{ij}^{r_2} - \mu 4^{r_2} AF^{r_2} - \sum_{g \in SA^{r_2}} \mu 5_g^{r_2} Q_g^{r_2} - \sum_{gh \in SAB^{r_2}} \mu 6_{gh}^{r_2} Q_{gh}^{r_2} \quad (17)$$

subject to constraints (2), (11), and (13).

Model (E)

Minimize

$$Z = \sum_{n \in N^{r_1}} \sum_{ij \in B^{nr_1}} T_{ij}^{nr_1} Y_{ij}^{nr_1} + \sum_{i \in SA^{r_1}} \left(F_i^{r_1} \times W_i^{r_1} \right) + \sum_{n \in N^{r_1}} \sum_{ij \in BF^{nr_1}} Y_{ij}^{nr_1} \left(V_i^{r_1} + V_j^{r_1} \right) + \sum_{ij \in IFF^{r_1}} \mu 7_{ij}^{r_1} \sum_{n \in N^{r_1}} Y_{ij}^{nr_1} + \sum_{ij \in FF^{r_1} \setminus IFF^{r_1}} \mu 8_{ij}^{r_1} \sum_{n \in N^{r_1}} Y_{ij}^{nr_1} + \sum_{ij \in FF^{r_1} \setminus IFF^{r_1}} \mu 8_{ij}^{r_1} \sum_{n \in N^{r_2}} Y_{ij}^{nr_2} + \sum_{m \in CCN} \sum_{ij \in \bigcup_{r_1} CCDA^{mr_1}} \mu 9_{ij}^{m} Y_{ij}^{mr_1} + \sum_{i \in SA^{r_1}} \mu 10_i^{r_1} \sum_{n \in N^{r_1}} \sum_{ij \in BF^{nr_1}} Y_{ij}^{nr_1} - \sum_{m \in CCN} \sum_{ij \in \bigcup_{r_1} CCDA^{mr_1}} \mu 9_{ij}^{m} D_{ij}^{m} \quad (18)$$

subject to constraints (3), (10), (12) and (14).

Model (F)

Minimize

$$Z = \sum_{n \in N^{r_2}} \sum_{ij \in B^{nr_2}} T_{ij}^{nr_2} Y_{ij}^{nr_2} + \sum_{i \in SA^{r_2}} \left(F_i^{r_2} \times W_i^{r_2} \right) + \sum_{n \in N^{r_2}} \sum_{ij \in BF^{nr_2}} Y_{ij}^{nr_2} \left(V_i^{r_2} + V_j^{r_2} \right) + \sum_{ij \in IFF^{r_2}} \mu 7_{ij}^{r_2} \sum_{n \in N^{r_2}} Y_{ij}^{nr_2} + \sum_{ij \in FF^{r_2} \setminus IFF^{r_2}} \mu 8_{ij}^{r_2} \sum_{n \in N^{r_2}} Y_{ij}^{nr_2} + \sum_{ij \in FF^{r_2} \setminus IFF^{r_2}} \mu 8_{ij}^{r_2} \sum_{n \in N^{r_1}} Y_{ij}^{nr_1} + \sum_{m \in CCN} \sum_{ij \in \bigcup_{q_1} CCDA^{mr_2}} \mu 9_{ij}^{m} Y_{ij}^{mr_2} + \sum_{i \in SA^{r_2}} \mu 10_i^{r_2} \sum_{n \in N^{r_2}} \sum_{ij \in BF^{nr_2}} Y_{ij}^{nr_2} - \sum_{m \in CCN} \sum_{ij \in \bigcup_{q_1} CCDA^{mr_2}} \mu 9_{ij}^{m} D_{ij}^{m} \quad (19)$$

subject to constraints (3), (10), (12) and (14).

Step 3 : Models (C), (D), (E), and (F) are close to pure network flow problems, and are also characterized as minimum cost network flow problems, which can be solved directly using the mathematical programming solver, CPLEX.

Step 4 : The lower bound of the optimal objective is obtained by summing up all four network costs.

B. The upper bound solution

The searching steps are listed below. We first define the symbols that are used in the heuristic:

$FFN, CFN, CFNS$: the fleet-flow network, the cargo-flow network, and the

Open Science Index, Mathematical and Computational Sciences Vol:4, No:6, 2010 publications.waset.org/10886/pdf

Open Science Index, Mathematical and Computational Sciences Vol:4, No:6, 2010 publications.waset.org/10886/pdf

cargo-flow networks;

$MFFNS, MCFNS$: the modified fleet-flow networks and the modified cargo-flow networks;

$fflow_i^a$, $fflow_i^b$: the fleet flows in the i[th] iteration, where superscript a (or b) indicates that the flows are infeasible (or feasible);

$cflow_i^a, cflow_i^b$: the cargo flows in the i[th] iteration, where superscript a (or b) indicates that the flows are infeasible (or feasible);

sol_i : the upper bound solution in the i[th] iteration, including both the cargo flows, $cflow_i^b$ and the fleet flows, $fflow_i^b$;

$\Delta cflow^a, \Delta cflow^b$: the increased flows for sol_i, where superscript a (or b), denotes that the increased flows do not (or do) assure that the cargo delivery volume is within the aircraft capacity;

$profit_i$: the objective value of sol_i;

$nfflow_{i+1}^b$, $ncflow_{i+1}^b$: the new feasible fleet and cargo flows in the $(i+1)^{th}$ iteration;

$nsol_{i+1}$: the new feasible upper bound solution in the $(i+1)^{th}$ iteration;

$nprofit_{i+1}$: the objective value obtained in the $(i+1)^{th}$ iteration;

The steps of upper bound are listed below:

Step 1 : Let the cargo flows for the initial lower bound solution be $cflow_1^a$. Solve $fflow_1^a$ based on $cflow_1^a$ as follows: first, construct a modified fleet-flow network ($MFFN$) which is the same as FFN (including the fleet size and other related constraints), except that each flight arc cost includes the original operating cost plus the sum of all profits obtained from the corresponding delivery arc flows of $cflow_1^a$. Note that the sum of the profits is set to be at most the aircraft capacity. Then, solve $MFFN$ to find $fflow_1^a$ using CPLEX;

Step 2 : Find $cflow_1^b$ based on $fflow_1^b$. We construct the modified cargo-flow networks ($MCFNS$) which are the same as CFNS, except that the cargo delivery arc flows in $MCFNS$ are restricted by the cargo loading constraints (7) and (8), based on $fflow_1^b$. Then, solve MCFNS to find $cflow_1^b$ using CPLEX;

Step 3 : Achieve an initial feasible solution sol_1, and its objective value $profit_1$ by combining $fflow_1^b$, and $cflow_1^b$. Set $i = 1$;

Step 4 : Solve $\Delta cflow^a$ based on sol_i to increase unserved cargos as follows. First, for every cargo delivery arc in each CFN, the flow upper bound is reset to be the capacity of the aircraft associated with the flight arc in $fflow_1^b$, minus its arc flow in $cflow_i^b$. For every demand arc in each CFN, recalculate the residual demand as the flow upper bound, which is equal to the projected cargo demand minus its arc flow in $cflow_i^b$. Then, solve $CFNS$ to find $\Delta cflow^a$ using CPLEX, based on the new flow upper bounds of the delivery and the demand arcs (all other parameters remain the same). Finally, in the same way update the flow upper bound for every delivery and demand arcs;

Step 5 : Add up $cflow_i^b$ and $\Delta cflow^a$ to form $cflow_{i+1}^a$, which, together with $fflow_i^b$, usually violates constraints (7) and (8);

Step 6 : Referring to Step 1, solve $fflow_{i+1}^b$ based on $cflow_{i+1}^a$;

Step 7 : Referring to Step 2, solve $cflow_{i+1}^b$ based on $fflow_{i+1}^b$;

Step 8 : By combining $fflow_{i+1}^b$ and $cflow_{i+1}^b$, we find the feasible solution, sol_{i+1}, and its objective value, $profit_{i+1}$;

Step 9 : If $profit_{i+1}$ is better than $profit_i$, then set $i = i+1$ and go to Step 4; else, go to Step 10;

Step 10 : Solve $\Delta cflow^b$ based on sol_{i+1} as follows. The method is similar to Step 4, except that the recalculation of the flow upper bound of every delivery arc is different. To increase the demand without violating constraints (7) and (8), we do not allow any flow to be augmented into the delivery arcs when the corresponding flight arc flows are zero. In other words, for every delivery arc, if the associated flight arc equals zero, then its flow upper bound is set as zero. Using the same technique as in Step 4, we can solve $\Delta cflow^b$;

Open Science Index, Mathematical and Computational Sciences Vol:4, No:6, 2010 publications.waset.org/10886/pdf

Step 11 : Add up $cflow^b_{i+1}$ and $\Delta cflow^b$ to form a new $cflow^b_{i+1}$ ($ncflow^b_{i+1}$);

Step 12 : Find a new $fflow^b_{i+1}$ ($nfflow^b_{i+1}$) based on $ncflow^b_{i+1}$. To do this, we first fix the $ncflow^b_{i+1}$ variables in the objective function (1), constraints (3), (7), (8), (9), (10), (12) and (14), and then solve the rest of Model (A). Finally, by combining $nfflow^b_{i+1}$ and $ncflow^b_{i+1}$, we find a new feasible solution and its objective value, that is, $nsol_{i+1}$ and $nprofit_{i+1}$;

Step 13 : If $nprofit_{i+1}$ is better than $profit_{i+1}$, then update sol_{i+1} and $profit_{i+1}$, and go to Step 10; else, we find the final feasible solution, sol_{i+1}.

C. The sub-gradient method and the solution process

Yan and Young[14]'s sub-gradient method for adjusting Lagrangian multipliers is applied in this study, so as to obtain good convergence in the iteration results. The steps of the Lagrangian relaxation-based algorithm is as follows:

Step 1 : Set iteration $k = 0$ and the initial Lagrangian multiplier μ^k to be 0;

Step 2 : Use CPLEX to solve Models (C), (D), (E) and (F) to get a lower bound $Z^L(\mu^k)$. If the solution is feasible and also satisfies the complementary slackness condition, then we have found an optimal solution and can stop the solution process. Otherwise, update the lower bound Z^L;

Step 3 : Apply the UP1 to find an upper bound $Z^U(\mu^k)$ 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, θ (i.e., $\left|\frac{(Z^U - Z^L)}{Z^L}\right| \le \theta$), or the computational time exceeds the computational time limit, stop the algorithm;

Step 5 : Adjust μ^k to help improve the convergence by applying the subgradient method developed in Yan and Young[14];

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

IV. Computational Tests

To evaluate the performance of the proposed solution algorithms, we performed numerical tests using operating data from two major Taiwan cargo airlines, with reasonable assumptions. We used the C computer language, coupled with the mathematical programming solver, CPLEX 9.0, to develop all the necessary programs. The tests were performed on a Pentium 4 – 3.2G with 1.5 Gb of RAM in the environment of Microsoft Windows XP. To preliminarily evaluate the exact solution methods for solving the test problems, we first used CPLEX to solve five problems (Southeast Asia, Asia, Asia-Europe, Asia-America, Global) with various sizes ranging from 8 cities, 4 B747-400F airplanes (100 metric tons each) for Airline r_1 and 10 cities, 6 MD-11F airplanes (80 metric tons each) for Airline r_2, to 62 cities, 19 B747-400F airplanes (100 metric tons each) for Airline r_1 and 42 cities, 17 MD-11F airplanes (80 metric tons each) for Airline r_2. All the cost parameters, cargo fare rates and other inputs, such as the flight times, the distance between two stations, the approved flight quota for each airport/airport pair, the available time slots at each airport and the ground handling time, for each airline, were primarily based on actual operating data. Note that the above inputs are adjustable to meet the operating requirements for other applications. In the tests, we set the convergence gap to be 0.03 and use CPLEX to solve the proposed model, Model A. The results are shown in Table 1. OBJ represents the objective function value obtained. LB represents the best objective function value of all the unexplored nodes in the branch-and-bound tree obtained by using CPLEX, which can serve as a lower bound of the problem. LBG (%) denotes the gap between OBJ and LB.

From Table 1, it can be seen that the running time has drastically increased as the problem scale increased. To save time, the computation was terminated when the running time exceeded 24 hours (86,400 seconds). As a result, except for Southeast Asia and Asia, other objective values could not be obtained within a reasonable time. As can be seen, as the problem scale increased, it was almost impossible to optimally solve a realistically large problem within a limited time.

TABLE I RESULTS FOR CPLEX FOR DIFFERENT PROBLEM SCALES

Problem scale	Southeast Asia	Asia	Asia-Europe	Asia-America	Global
Variables	101,538	425,874	1,121,382	1,493,658	2,254,361
Constraints	20,936	65,256	146,848	168,652	234,286
OBJ (1,000NT$)	-57,319	-174,860	-225,694*	-295,843*	-359,668*

LB (1,000NT$)	-58,956	-180,025	-308,612	-413,388	-583,894
LBG (%)	2.856	2.954	36.739*	39.732*	62.342*
Time (seconds)	5,038.79	40,288.54	88,127.58	87,532.14	88,559.67

*: The solution could not converge within the convergence gap in less than 24 hours (86,400 sec).

To evaluate the performance of the solution algorithms in solving coordinated fleet routing and flight scheduling problem, we first use the solution algorithm to solve the problem proposed in Yan and Chen[8]. Before implementing the tests, the following parameters are set: in the Lagrangian based algorithm, the convergence tolerance θ, is set to be 0.03. All the problems are solved to within a convergence tolerance (θ) of 0.03. The results are shown in Table 2. CVG (%) denotes the gap between OBJ and the lower bound of the solution algorithm itself. For ease of comparison, the OBJs were obtained by our solution algorithm within a limited computational time of 12 hours (43,200 seconds).

As shown in Table 2, the OBJ obtained by the solution algorithm was better than that obtained by CPLEX. Moreover, the LBG of the solution algorithm was also better than that of CPLEX. In particular, for the Lagrangian based algorithm, compared with CPLEX, the LBG was improved from 2.856% to 2.592%. As for the computation time, CPLEX has the longest, Lagrangian based algorithm the shortest. The test results show that the proposed Lagrangian based algorithm perform better than CPLEX. In short, the solution algorithm is improvement over CPLEX, having the potential to solve middle/large scale coordinated fleet routing and flight scheduling problems.

TABLE II TEST RESULTS FOR SOUTHEAST ASIA

	CPLEX		Lagrangian based algorithm	
	Airline r_1	Airline r_2	Airline r_1	Airline r_2
OBJ (1,000 NT$)	-57,319		-57,466	
	-27,530	-29,789	-27,613	-29,853
LBG (%)	2.856		2.592	
CVG (%)	2.856		2.622	
Computation time (sec)	5038.79		2515.47	
#Iterations	--		296	
Fleet size	4	6	4	6
Frequency (flights/week)	33	40	33	40
Service rate (%)	99.598	99.672	99.614	99.677
Average load factor (%)	68.01	68.92	68.04	68.945

-- Not available.

V. CONCLUSIONS

We develop the Lagrangian based algorithm for solving coordinated fleet routing and flight scheduling problems. The Lagrangian based algorithm is designed mainly based on the Lagrangian relaxation method, a subgradient method, and a heuristic for finding the upper bound solution. To show how well the solution algorithms can be applied in real practices, we perform a case study utilizing real operating data from two Taiwan airlines. The objective value obtained by the Lagrangian based algorithm is better than that obtained with CPLEX, by 0.256%. To sum up, the proposed Lagrangian based algorithms performed well and could be useful for allied airlines to solve coordinated fleet routing and flight scheduling problems.

ACKNOWLEDGEMENTS

This research was supported by a grant (NSC 95-2420-H-008-001D-DR) from the National Science Council of Taiwan. We also thank the two airlines for providing the test data as well as their valuable opinions.

REFERENCES

[1] Etschmaier M, Mathaisel D (1984) Aircraft Scheduling: The State of the Art. AGIFORS XXIV 181-209
[2] Yan S and Tseng C H (2002) A Passenger Demand Based Model for Airline Flight Scheduling and Fleet Routing. Computers and Operations Research 29: 1559-1581
[3] Barnhart C, Kniker T, and Lohatepanont M (2002) Itinerary-Based Airline Fleet Assignment. Transportation Science 36(2): 199-217
[4] Lohatepanont M and Barnhart C (2004) Airline Schedule Planning: Integrated Models and Algorithms for Schedule Design and Fleet Assignment. Transportation Science 38(1): 19-32
[5] Yan S, Tang C H and Lee M C (2007) A Flight Scheduling Model for Taiwan Airlines under Market Competitions. OMEGA - International Journal of Management Science 35: 61-74
[6] Yan S, Chen S C and Chen C H (2006) Fleet Routing and Timetable Setting with Multiple Timeliness Air Cargo's Demand. Transportation Research 42E(5): 409-430
[7] Yan S and Chen C H (2007) Coordinated Scheduling Models for Allied Airlines. Transportation Research 15C: 246-264
[8] Yan S and Chen C H (2008) Optimal Flight Scheduling Models for Cargo Airlines under Alliances. Journal of Scheduling 11(3): 175-186
[9] Garey M R and Johnson D S (1979) Computers and intractability: A Guide to the Theory of NP-Completeness, W.H. Freemean & Company, San Francisco
[10] Lee B C (1986) Routing Problem with Service Choices, Flight Transportation Laboratory Report R86-4, MIT, Cambridge, Massachusetts
[11] Teodorovic D (1988) Airline Operation Research. Gordon and Breach Science, New York
[12] Ball M O, Magnanti T L, Monma C L and Nemhauser G L (1995) Network Routing. Handbooks in Operations Research and Management Science
[13] Fisher M L (1981) The Lagrangian Relaxation Method for Solving Integer Programming Problems. Management Science 27: 1-18
[14] Yan S and Young H F (1996) A Decision Support Framework for Multi-Fleet Routing and Multi-Stop Flight Scheduling. Transportation Research 30A: 379-398
[15] Camerini P K, Fratta L and Maffioli F (1975) On Improving Relaxation Methods by Modified Gradient Techniques. Mathematical Programming Study 3: 6-25
[16] Barnhart C, Johnson E D, Nemhauser G L, Savelsbergh M W P and Vance P H (1998) Branch and Price Column Generation for Solving Hugh Integer Programs. Operations Research 46: 316-329

Open Science Index, Mathematical and Computational Sciences Vol:4, No:6, 2010 publications.waset.org/10886/pdf