Available online at www.sciencedirect.com

ScienceDirect

Transportation Research Part C 15 (2007) 235–245

TRANSPORTATION RESEARCH PART C

www.elsevier.com/locate/trc

A Lagrangian relaxation based heuristic for the consolidation problem of airfreight forwarders

Kuancheng Huang [a,*], Wenhou Chi [b]

[a] *Department of Transportation Technology and Management, National Chiao Tung University, No. 1001, Ta Hsueh Road, Hsinchu City 300, Taiwan*

[b] *Asia Pacific Distribution Center, Logitech Far East Ltd., No. 2 Creation Road IV, Science-Based Industrial Park, Hsinchu City 300, Taiwan*

Received 28 January 2006; received in revised form 2 July 2006; accepted 28 August 2006

Abstract

International air cargo is an operation-intensive industry, involving complex procedures and many players. As an important player, airfreight forwarders need to consolidate the collected goods skillfully in order to satisfy the requirements of the shippers and, at the same time minimize the expense charged by the airlines. However, the air cargo rate structure is very complicated, making the consolidation a difficult mixed-integer programming problem for the airfreight forwarder. In this paper, the consolidation problem is first transformed into a well-known set covering problem by treating a feasible consolidated shipment as a set. Lagrangian Relaxation is used as the backbone to develop a recursive heuristic algorithm. Based on the numerical experiment, the heuristic algorithm generates solutions very close to optimality. In particular, a sensitivity analysis is performed with respect to the degree of concavity. The results suggest that the solution algorithm can be used as a core module of the decision support system for air cargo consolidation.

Keywords: Concave minimization; Set covering problem; Lagrangian relaxation; Freight consolidation; Air cargo; Decision support

1. Introduction

Due to world trade liberalization and global logistics operation, the air cargo industry has been booming for the past decade, and recent forecasts project a very promising development for the next two decades. For example, one aircraft manufacturer (Boeing, 2004) estimates that the average yearly growth rate is as high as 6.2%. In particular, a remarkable traffic volume increase is expected for the Asia-Pacific region, due to the global logistics operation of high-tech manufacturing firms in the region. The growth rates for traffic related to Asian markets will exceed the world average, and the share of world-wide air cargo traffic for the region will increase from 47.6% to 59.4% by 2023. However, given these promising projections, it will be a challenge to

* Corresponding author. Tel.: +886 3 5731928; fax: +886 3 5720844.
E-mail address: kchuang@cc.nctu.edu.tw (K. Huang).

0968-090X/$ - see front matter
doi:10.1016/j.trc.2006.08.006

raise the efficiency of the industry to meet this demand. International air cargo is an operation-intensive industry, which involves complex procedures and many players. Airfreight forwarders play an important role in the whole process. They are the air service providers to the shippers as well as the consumers to the air airlines.

Airfreight forwarders need to consolidate the collected shipments skillfully in order to satisfy the requirement of the shippers, and at the same time minimize the expense charged by the airlines. In general, local airfreight forwarders make the consolidation arrangements manually based on their experience. However, the air cargo rate structure is very complicated. It takes into account both the weight and the volume of the shipments. In addition, the airlines usually provide a significant quantity discount. These characteristics make the airfreight forwarder consolidation problem (AFCP) of this study a very difficult mathematical programming problem. Past studies have applied mixed integer programming (MIP) to model this problem. However, owing to the inherent computational load of MIP, it is hard to solve large-scale problems within a practical time.

The goal of this study is to develop a suitable algorithm as the core module of the decision support system for air cargo consolidation. The AFCP is first transformed into a well-known set covering problem (SCP) by treating a feasible consolidated shipment as a set. The Lagrangian Relaxation, a successful approach for SCP reported in prior researches, is used as the backbone to develop a recursive heuristic algorithm. However, the space of the potential sets in the SCP is huge, as there are numerous ways to consolidate the collected cargo items. Thus, the solution space, consisting of a limited number of permissible consolidated shipments, is initially generated and adjusted through the iterative solution procedure. In addition, to raise the computational efficiency and solution quality, technical issues (such as feasible solution realization, Lagrangian multiplier update, and solution space adjustment) are examined carefully, and the solution algorithm is tuned accordingly. Finally, a series of numerical experiments are performed to verify the solution algorithm.

This paper is organized as follows: the next section provides the background of the problem, and in particular elaborates upon the special features of the air cargo rating system. In addition, the MIP model of the AFCP is presented, and related previous researches are reviewed. In Section 3, the development of the solution algorithm is explained in detail, including the transformation of an AFCP into an SCP, the procedures in the recursive algorithm, and the stopping criteria. The numerical experiment and the results of the sensitivity analysis are described in Section 4. Finally, the findings of this study are concluded in Section 5.

2. Problem backgrounds and prior researches

2.1. Characteristics of the air freight forwarder consolidation problem

Due to the mode characteristics of air transportation, the air cargo rating system takes into account the volume of the shipment, in addition to its gross weight. The so-called volume weight is derived by dividing the volume in cm^3 by a constant $6000\,cm^3/kg$, currently adopted by the industry. The chargeable weight is then the greater of the gross weight and the volume weight. Therefore, there is an incentive for airfreight forwarders to combine the air cargo items, later referred to as *items*, from different shippers into a consolidated shipment, as the overall chargeable weight of a consolidated shipment with low-density items and high-density items is less than the sum of the chargeable weights of the individual items.

Airlines usually offer quantity discount to air freight forwarders based on the chargeable weight described above. The cost curve in Fig. 1 is a numerical example, in which the rates are \$30/kg, \$20/kg, and \$18/kg for the shipments less than 45 kg, less than 100 kg, and greater than 100 kg, respectively. However, in order to keep the cost curve remain monotonously non-decreasing, shipments between 30 kg and 45 kg are charged as 45 kg for \$20/kg, causing the marginal cost within the range to be zero. Similarly, shipments greater than 90 kg are charged as 100 kg for \$18/kg. The weights such as 30 kg and 90 kg in this example are called the weight breaks in the air cargo industry. Finally, for the purpose of mathematical modeling, the weight ranges in terms of marginal cost are later referred to as *rate segments*. For example, there are five segments in this example.

Quantity discount provides another incentive for air freight forwarders to consolidate the shipments. However, given multiple available flights with possibly different cost curves and multiple items with various gross weights and volume weights, air freight forwarders must consolidate the items very carefully to minimize the expenses charged by the airlines while satisfying the shippers' requirement and preferences.

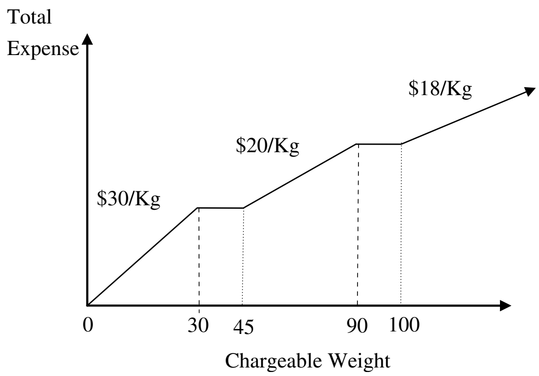

Fig. 1. Numerical example for quantity discount of the air cargo rating system.

2.2. Mathematical programming model

There have been some researches regarding the operation of air freight forwarders. For example, Jaeger (1976) dealt with the airfreight consolidation problem in the United States with a cost curve similar to the one in Fig. 1. In the formulation, routing decisions are incorporated, but the conversion of volume weights is not considered. To cope with the operational decisions of air freight forwarders in Hong Kong, Xue and Lai (1997) developed a container rental and cargo shipment problem by MIP formulation. Wu (2003) developed the following MIP model to incorporate the problem features described in the previous subsection.

- n: maximum number of items;
- i: indices of items ($i = 1, \ldots, n$, assuming I is the set of all items);
- m: maximum number of flights;
- j: indices of flights ($j = 1, \ldots, m$, assuming J is the set of all flights);
- M_i: set of all flights allowed to deliver items i ($M_i \subseteq J$, $\forall i$);
- l: maximum number of rate segments;
- k: indices of rate segments ($k = 1, \ldots, l$, assuming K is the set of all rate segments);
- R_{jk}: rate of segment k for flight j;
- G_i: gross weight of item i;
- V_i: volume weight of item i;
- X_{jk}: ending point of segment k for flight j; where $X_{j0} = 0$, $\forall j$;
- z_{ij}: binary decision variable representing item i is assigned to flight j;
- y_{jk}: decision variable representing the chargeable weight of segment k for flight j;
- w_{jk}: binary decision variable representing y_{jk} reaches the maximum value

$$\text{Min} \sum_{j=1}^{m} \sum_{k=1}^{l} R_{jk} \times y_{jk} \tag{1}$$

$$\sum_{j \in M_i} z_{ij} = 1 \quad \forall i \in I \tag{2}$$

$$\sum_{k=1}^{l} y_{jk} - \sum_{i=1}^{n} z_{ij} \times G_i \geqslant 0 \quad \forall j \in J \tag{3}$$

$$\sum_{k=1}^{l} y_{jk} - \sum_{i=1}^{n} z_{ij} \times V_i \geqslant 0 \quad \forall j \in J \tag{4}$$

$$y_{jk} - w_{j(k-1)} \times (X_{jk} - X_{j(k-1)}) \leqslant 0 \quad \forall j \in J, \ \forall k = K - \{1\} \tag{5}$$

$$y_{j1} - (X_{j1} - X_{j0}) \leqslant 0 \quad \forall j \in J \tag{6}$$

$$y_{jk} - w_{jk} \times (X_{jk} - X_{j(k-1)}) \geqslant 0 \quad \forall j \in J, \ \forall k \in K \tag{7}$$

$$z_{ij}, w_{jk} \quad \text{binary} \tag{8}$$

$$y_{jk} \geqslant 0 \tag{9}$$

The objective function (1) is to minimize the total expense, which is derived by adding the charges for all segments of all flights (weight y_{jk} multiplied by rate R_{jk}). In particular, some of the R_{jk}'s are zero to reflect the flat (zero marginal cost) region of the cost curve as shown in Fig. 1. Constraint (2), with the binary variable z_{ij}, specifies that each item i must be assigned to one of the qualified flights represented by the set M_i. Constraint (3) and Constraint (4) ensure that, for each flight, the chargeable weight is the greater of the gross weight and the volume weight of the consolidated shipment. Constraint (5) regulates that the segment weight (y_{jk}) must be within the segment range ($X_{jk} - X_{j(k-1)}$) on the condition that the weight of the previous segment has reached the maximum, i.e., $w_{j(k-1)} = 1$, otherwise the segment weight has to be zero. Constraint (6) is a special case of the first segment for Constraint (5). Constraint (7) is used to justify the meaning of the binary variable w_{jk}. As long as the segment weight has reached the maximum, i.e., $y_{jk} = X_{jk} - X_{j(k-1)}$, the binary decision variable w_{jk} is free to be 1. In general, the relation $w_{jk}(X_{jk} - X_{j(k-1)}) \leqq y_{jk} \leqq w_{j(k-1)}(X_{jk} - X_{j(k-1)})$ should hold for each segment based on Constraint (5) and Constraint (7). These constraints ensures that the decision variables representing the segment weights (y_{jk}'s) follow the cost curve, which includes the segments with generally decreasing marginal cost and one flat (zero-marginal cost) segment between them. Finally, Constraint (8) describes that the item-flight assignments z_{ij}'s and the maximum segment volume indicators w_{jk}'s are binary decision variables, and Constraint (9) describes that the segment weights y_{jk}'s are non-negative real decision variables.

2.3. Prior related researches

The above formulation is a difficult MIP problem. In general, there are two approaches to solve this kind of problem, exact-solution algorithms and heuristics. For the former, branch-and-bound (such as Horst and Thoai, 1998) and dynamic programming (such as Ward, 1999) are typical examples. This approach usually involves a high computational load and is applicable to small-scale problems. For example, in Wu (2003), with the optimization package LINGO, the AFCP as (1)–(9) can be solved within a reasonable operational time if the problem size is smaller than the assignment of 5 flights and 60 items.

On the other hand, heuristics greatly alleviate the computational load and are more suitable for the problems with practical size, though the solution generated is not guaranteed to be optimal. There have been various kinds of methods that were heuristically developed to handle different kinds of problems. For example, Jordan (1986) applied successive marginal cost approximation to model a concave cost curve. Both Larsson et al. (1994) and Amiri and Pirkul (1997) made use of Lagrangian relaxation to design a solution algorithm. In addition, some researches tried combinatorial optimization techniques, such as the tabu search in Yan and Luo (1998). For a summary of solution algorithms, Horst et al. (2000) serve as an excellent reference.

Instead of solving the problem directly, many researches have tried to transform the problem into some other well-known problem. For example, some scheduling problems (e.g., the airline crew scheduling problem) and routing problems (e.g., the vehicle dispatching problem) have been transformed to the classical set covering problem (SCP). For example, Irnich (2000) described how a multiple-location pickup-delivery problem is transformed into a SCP before the solution is derived. The key benefit of this transformation is to take advantage of the solving techniques of the well-studied classical SCP such as the Lagrangian relaxation in Caprara et al. (1999). The main theme of the research in the present paper follows this concept. The whole solution algorithm is presented in the following section.

3. Development of the solution algorithm

To illustrate the development of the solution algorithm, several key design concepts are expanded upon, including the transformation of the AFCP to an SCP-type problem, the Lagrangian-relaxed solution, and the adjustment of the solution space. At the end, the stopping criterion of the recursive algorithm is discussed, and a summary of the solution algorithm is presented.

3.1. Transformation of the consolidation problem into the set covering problem

By treating a consolidated shipment, later referred to as a *consolidation combination* or simply a *combination*, for a specific flight as a set, the AFCP of (1)–(9) can be transformed into a problem very similar to a classical SCP as follows:

- i: indices of items ($i = 1, \ldots, n$, assuming I is the set of all items);
- k: indices of consolidation combinations (assuming K is the set of all possible combinations);
- N_j: set of all permissible consolidation combinations for flight j;
- C_k: cost of combination k;
- a_{ik}: binary constant; $a_{ik} = 1$ if item i is in the combination k, and $a_{ik} = 0$ if otherwise;
- x_k: binary decision variable representing that combination k is selected

$$\text{Min} \sum_{k \in K} C_k x_k \tag{10}$$

$$\sum_{k \in K} a_{ik} x_k \geqslant 1 \quad \forall i \in I \tag{11}$$

$$\sum_{k \in N_j} x_k \leqslant 1 \quad \forall j \in J \tag{12}$$

$$x_k, \quad \text{binary} \tag{13}$$

The objective function (10) is to minimize the total expense by combining the costs of the selected combinations. Constraint (11) ensures that each item is incorporated in at least one of the selected combinations. Especially as the set covering problem (SCP) is generally easier to solve than the set partitioning problem (SPP), the greater-or-equal-to sign is used in (11), instead of the equal sign. As it is thus possible that one item can be included in multiple selected combinations, extra steps must be taken to eliminate redundancy in the solution algorithm so as to correct this situation. Eq. (12) is not a typical constraint in a classical SCP. It requires that at most one consolidation combination can be selected for each flight. Technically, multiple shipments from one forwarder can be carried by a single flight. However, given the typical cost curve in Fig. 1, the overall cost is lower if they are combined as one big shipment. In addition, limiting one combination for each flight can ensure the weight of the consolidation shipment does not exceed the capacity allocated to a forwarder. Finally, in Constraint (13), the binary variable x_k represents the choice of the consolidation combinations.

3.2. Determination of the Lagrangian relaxed solution

The above SCP-type problem (10)–(13) can be re-formulated, similar to the approach in Caprara et al. (1999), by relaxing Constraint (11) to derive the following Lagrangian relaxed problem. As Constraint (11) can be written as $1 - \sum a_{ik} x_k \leqslant 0$, the associated Lagrangian multipliers needs to be greater than or equal to zero, given the minimization objective function. As for the initial value of the Lagrangian multiplier for each item, it is set as the cost if the item were shipped individually on the least expensive permissible flight. The values of the Lagrangian multipliers are updated, as described in Section 3.3, and become smaller through iterations as good combinations are formed to take into account the benefit of consolidation.

- u_i: Lagrangian multiplier for item i, where $u_i \geqslant 0\ \forall i$; $\boldsymbol{u}$ is the column vector consisting of all Lagrangian multipliers;
- I_k: set of items covered by combination k, i.e., $I_k = \{i \in I\text{: } a_{ik} = 1\}$

$$L(\boldsymbol{u}) = \text{Min} \sum_{k \in K} c_k(\boldsymbol{u}) x_k + \sum_{i \in I} u_i \tag{14}$$

$$\sum_{k \in N_j} x_k \leqslant 1 \quad \forall j \in J \tag{15}$$

$$x_k, \quad \text{binary} \tag{16}$$

$$\text{where} \quad c_k(\boldsymbol{u}) = C_k - \sum_{i \in I_k} u_i \quad \forall k \in K \tag{17}$$

This Lagrangian relaxed problem is very similar to a typical Lagrangian relaxed SCP as in Caprara et al. (1999), except for Constraint (15). If only the objective function (14) and Constraint (16) are considered, then all the combinations with negative $c_k(\boldsymbol{u})$ are selected, i.e., $x_k = 1$ for $c_k(\boldsymbol{u}) \leqslant 0$. However, to take Constraint (15) into account, only the most negative one is selected if multiple combinations are with negative $c_k(\boldsymbol{u})$. In terms of computational consideration, this decision rule can be implemented by sorting the values of $c_k(\boldsymbol{u})$'s for all combinations. Conceptually, the value of $c_k(\boldsymbol{u})$ can be viewed as how *well* combination k is made. The more negative the value is, the better the combination is. Based on Eq. (17), the value of $c_k(\boldsymbol{u})$ is computed by subtracting the sum of the associated Lagrangian multipliers from the expense charged for combination k. It is worth noting that the Lagrangian multipliers are related to the dual variables. If they are interpreted as the fair prices for delivering the items individually in the open market, the difference in (17) can be thought of as the cost saving for choosing a specific combination. Therefore, the more negative the value is, the more desirable the combination is.

One alternative way to relax the problem (10)–(13) is to focus the special Constraint (12), which allows at most one set to be chosen for each flight. The resulted problem turns out to be a typical SCP with modified set costs. However, SCP is a relatively difficult integer programming problem when compared to above relaxed problem, which can be solved by a simple sorting procedure. In addition, as the ranked list from the sorting procedure represents how desirable the combinations are, it is in fact re-used in adjusting the solution space, as described in Section 3.4, to fully take advantage of the computational effort in the sorting procedure.

3.3. Realization of the feasible solution and updating of the Lagrangian multipliers

As Constraint (11) is relaxed, the Lagrangian relaxed solution derived from the previous subsection is generally infeasible as some items may not be covered by the selected combinations. In general, the feasible solution in a Lagrangian relaxation based approach can be derived from the relaxed solution. In finding such a feasible solution, the degradation in the objective function value is supposed to be minimized. However, the procedure to obtain the feasible solution should not be too complicated so as not to result in a heavy computational load. This study proposes a relatively simple procedure, which tries to replace one selected combination by another unselected combination. Given the ranked list of the combinations derived by sorting the values of $c_k(\boldsymbol{u})$'s, the unselected combination with the least increase for the objective function value across all flights is examined first. The examined combination is introduced to replace the selected combination of the same flight, if the covered items remained covered, and at least one more uncovered item becomes covered. Otherwise, this combination is discarded, and the next combination causing the least increase of the objective function value in the list is considered. The procedure is terminated if the modified solution becomes feasible, or if there is no unselected combination left.

In the above procedure, a feasible solution is not guaranteed. However, to reduce computation complexity, an unselected combination is discarded if it does not pass the examination. Thus, the examination for each unselected combination can only possibly be performed once, and the number of possible examinations is at most equal to the number of combinations. As described in the next subsection, the number of combinations is limited, and thus the computational load is significantly reduced. To cope with the problem of failing to generate a feasible solution, the incumbent best solution is recorded across iterations in the iterative solution algorithm.

The objective function value of the feasible solution and that of the Lagrangian relaxed problem are denoted by B and $L(\boldsymbol{u})$, respectively, and they are used in updating the Lagrangian multipliers according to the following equations (Held and Karp, 1970).

- t: index for iteration;
- K_i: set of combinations covering item i, i.e., $K_i = \{k \in K\colon a_{ik} = 1\}$;
- $s(\boldsymbol{u})$: vector used to update Lagrangian multipliers, where $s_i(\boldsymbol{u})$ is the ith element of the vector for item i

$$s_i(\boldsymbol{u}^t) = 1 - \sum_{k \in K_i} x_k(\boldsymbol{u}^t) \quad \forall i \in I \tag{18}$$

$$u_i^{t+1} = \max\left\{u_i^t + \lambda \frac{B^t - L(\boldsymbol{u}^t)}{\|s(\boldsymbol{u}^t)\|^2} s_i(\boldsymbol{u}^t), 0\right\} \quad \forall i \in I \tag{19}$$

The direction for updating the Lagrangian multipliers is computed as in (18), where $x_k(\boldsymbol{u}^t)$ is the Lagrangian relaxed solution of (14)–(17) for the current iteration. Given the direction, the step size is determined as in (19) by taking into account the gap between the objective function values of the feasible solution and the relaxed solution, the magnitude of the direction vector, and a pre-determined constant λ, which can also be adjusted dynamically based on problem characteristics. For the numerical experiment described later, it is chosen as 0.1. In addition, Lagrangian multipliers are forced to be greater or equal to zero in the process.

3.4. Adjustment of solution space

Although the AFCP can be transformed into an SCP-type problem, the number of possible combinations is tremendous when the problem size is increased. For example, for the problem with 5 flights and 20 items, the number of possible combinations is up to 5×2^{20}. Therefore, instead of generating all the possible combinations, only a limited number of combinations are generated initially. This partial set of possible combinations, later referred to as the *solution space*, is adjusted in the recursive solution algorithm based on the information derived in each iteration. The adjustment procedure is designed to maintain a list of *good* combinations, which serves as a basis to produce a solution with good quality. In the following sub-subsections, the generation of the initial solution space is explained first. Then, the steps for removing the poor combinations from the list are described. Finally, the ideas for finding the potentially well-arranged combinations are presented.

3.4.1. Introduction of the initial solution space

To generate a good feasible solution in the initial solution space, all the items are sorted according to their densities. Each item in this sorted list is sequentially assigned to a flight, chosen in a manner from $j = 1$ (the first flight) to $j = m$ (the last flight) and then moving backward from $j = m$ to $j = 1$. However, the limitation and preference for item-flight assignment is taken into consideration. If there is a prohibited item-flight assignment, the specific flight is skipped for one time. In addition, if a flight has reached its capacity, it is excluded from further assignment. The goal of this procedure is to generate initial permissible consolidation combinations with a relatively good mix of high- and low-density items. If a feasible solution is not generated from this process, some post-processing is required to derive a feasible solution contained the initial solution space. Finally, in order not to be confined by the initial solution, for each flight, the combinations consisting of each one item and the combinations consisting of the maximum number of items are also included in the initial solution space.

3.4.2. Removal of non-promising combinations

As explained earlier, conceptually the value of $c_k(\boldsymbol{u})$ can be viewed as an indicator or a *score* of how desirable the combination k is. Through iterations, the score may change as the Lagrangian multipliers are updated constantly. To acquire a general estimation of the desirability of a combination, a concept similar to exponential smoothing for time series forecast is applied (Neter et al., 1993, p. 807). Given a pre-determined constant α, the old smoothed score is multiplied by $(1 - \alpha)$ and then added to the newly derived $c_k(\boldsymbol{u})$ multiplied by α for each iteration. (For the numerical experiment described later, it is chosen as 0.2.) This new smoothed score is used as the yardstick to determine whether a combination should be kept in the solution space. After sorting these scores, only a limited number of combinations (e.g., 20 per flight in the following numerical experiment) with top-ranking scores *survive* for the next iteration, and the rest of the combinations are removed from the solutions space to keep the computation tractable.

3.4.3. Generation of promising combinations

The surviving combinations serve as the basis to generate new promising combinations by attempting to add one more specific item to, or removing a specific item from the surviving combinations. Based on Eq. (18), the item with the largest $s_i(\boldsymbol{u})$ is chosen as the *adding item*. Similarly, the one with the smallest $s_i(\boldsymbol{u})$ is chosen as the *subtracting item*. As defined in Eq. (18), $s(\boldsymbol{u})$ is not only interpreted as the deviation from the equality in the relaxed Constraint (11), it is also the vector used in updating the Lagrangian multipliers in Eq. (19). Therefore there is a better chance of generating a combination with more negative $c_k(\boldsymbol{u})$ by adding an item with a high $s_i(\boldsymbol{u})$ according to Eqs. (19) and (17). Likewise, if an item with a low $s_i(\boldsymbol{u})$ is removed from the existing combinations, the newly generated one is more likely to have a more negative $c_k(\boldsymbol{u})$. Particularly, while determining the adding/subtracting items, ties are broken randomly to increase the chance of generating potentially good combinations.

Since these combinations have been sorted based on the value of $c_k(\boldsymbol{u})$ while solving the Lagrangian relaxed problem, an attempt to generate new combinations can easily be carried out from the top of the ranked list. For each flight, this attempt stops if the number of total combinations exceeds a pre-determined limit. One reason for limiting the number of new combinations generated is to reduce the computational load. An equally important reason is that it is less likely to generate good combinations when based on the combinations at the bottom of the list.

3.5. Stopping criteria and a summary of the solution algorithm

The whole solution algorithm is summarized as a flowchart in Fig. 2. The stopping criteria in particular needs to be further addressed. Generally speaking, the solution algorithm of a Lagrangian relaxation problem terminates if the upper bound from the feasible solution and the lower bound from the relaxed problem are identical. When this does not happen, the algorithm usually stops if it is found that the gap between the upper bound and the lower bound is within an acceptable value, which is usually relatively small.

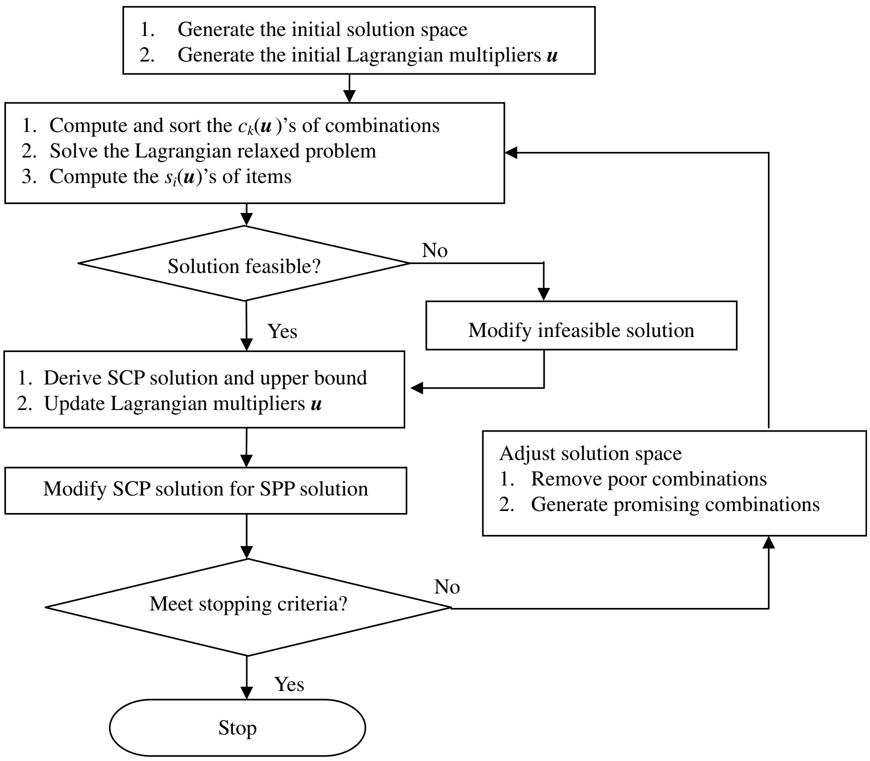

Fig. 2. Flowchart of the solution algorithm.

Based on the key ideas of the solution algorithm described in the previous sub-sections, the lower bound is only valid for the SCP problem in each iteration, and it is not valid for the original consolidation problem. The reason for this is that the solution space only contains a partial set of possible combinations. Thus, the stopping criteria must be adjusted even though the upper bound remains valid.

This study chooses to stop the iterative solution algorithm based on the number of iterations performed, similar to the criterion used in some tabu search algorithms, such as Taillard (1993). In the numerical experiment presented as an example in the next section, the stopping criterion is set at 500–900 iterations. Generally speaking, the number of iterations required is increased slightly for a larger size problem.

4. Numerical experiment

The test problems of the numerical experiment are designed based on randomly generated cargo item information. They are solved by the heuristic algorithm described in the previous section, and the objective function value derived is compared with that of the optimal solution, which is obtained by using the LINGO optimization package to solve the AFCPs (1)–(9).

To vary the problem size, the number of items is set to be 20, 30, 40, 50, and 60 with the corresponding number of flights set to be 2, 3, 4, 5, and 6. For each problem size, the gross weights and volume weights for 10 groups of items are created by the random number generator in the EXCEL package with the parameters specified in Table 1. The range for the densities of the items is between 0.3 to 3.0, and the overall density of all items is close to 1. Assuming the rate structures are the same for all flights, there are two levels of quantity discount. The rate differences are $2 and $5, respectively, as shown in Table 2. Finally, no limitation is imposed on the item-flight assignment, and the capacities for all flights are 1500 kg.

The solution quality is presented in terms of percentage for the heuristic solution with respect to the optimal solution. The average and the worst case of the 10 test problems for each problem size are summarized in Fig. 3. When the rate difference is as small as $2, the average deviations from optimality for all the problem sizes are within 0.5%, and, even for the worst case, the maximum gaps are all less than 1%. The average gap for all 50 test problems is 0.29%, and it appears that the solution quality is not sensitive to problem size.

When the rate difference is increased to $5, the average gap for all the problem sizes are within 1.0%, and, for the worst case, the maximum gap is about 2%. The average gap for all 50 test problems is increased to 0.50%, and it still appears that the solution quality is not sensitive to problem size. As the level of quantity discount increases, the degree of concavity of the cost functions increases. As expected, the solution quality is degraded, but appears to be acceptable. Especially, from the viewpoint of practical operation, a $5 rate difference is pretty high for most air cargo markets.

As for the analysis of the solution time, both the heuristic algorithm and the MIP models by LINGO were executed by the same PC (Pentium III, 1.06 GHz). The execution time of the heuristic algorithm, coded by MATLAB, is very consistent for all the test problems. It ranges from 6 s to 35 s, roughly increasing in a linear fashion with respect to problem size. The computational time mainly depends on the number of iterations and

Table 1
Parameters of random number generator for item information

Distribution type	Upper bound	Lower bound	Average	Number of test groups
Uniform	56.25	168.75	112.5	10

Table 2
Quantity discount for each flight

Rate difference	Weight (kg)					
	0–100	100–300	300–500	500–700	700–1000	1000–1500
$2	90	88	86	84	82	80
$5	90	85	80	75	70	65

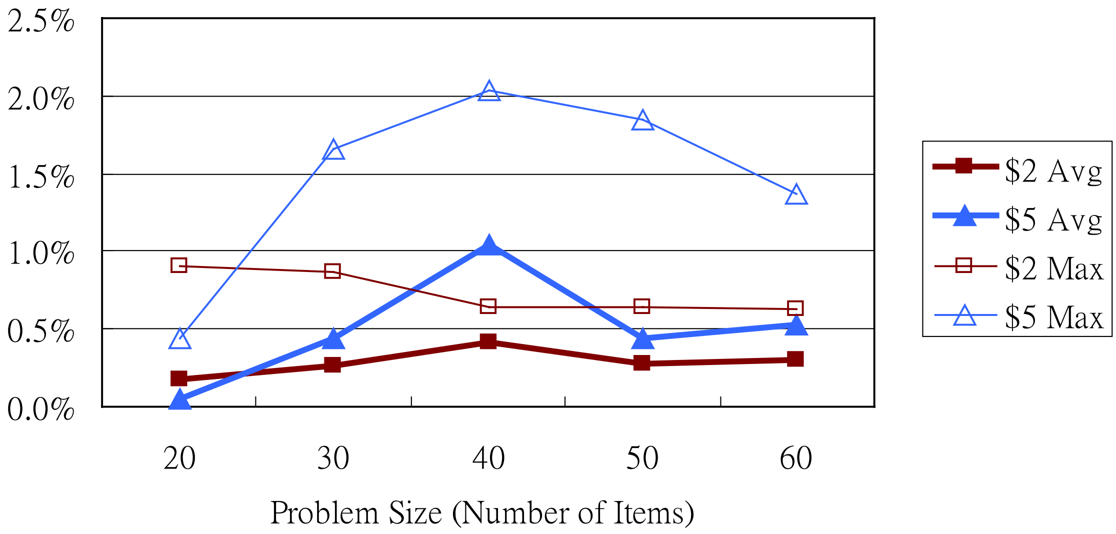

Fig. 3. Solution quality of the solution algorithm.

the size of the solution space, or equivalently the number of allowed combinations through iterations. For the solution time of the MIP model by LINGO, it is insignificant (from fractional second to a few seconds) for all test problems, except for the problem size of 60 items and 6 flights, in which there are 426 binary variables (499 variables in total) and 212 constraints. Although the branch-and-bound procedure of LINGO terminates quickly for most instances of this scale, it takes more than 1 min for 5 out of the 20 test problems. The longest computation time is 10.5 min, and the average computation time is about 54 s for these largest test problems.

The analysis of the solution time suggests that the standard optimization software package might serve the purpose well for the problem scale similar to those tested in the numerical experiment. However, the solution time can be highly variable. Besides, as the computation time for the branch-and-bound procedure can increase in an exponential way, the heuristic algorithm should be more suitable for larger problems. In particular, given the dynamic business environment, shippers nowadays tend to make small but frequent shipments. Together with the effect of the merges among the airfreight forwarders, the number of items handled by a forwarder increases significantly. At the same time, the number of available flights increases as well, given the steady growth of the air transportation market. It can be expected that the size of the AFCPs is likely to become bigger and bigger, and there should be a need of a heuristic algorithm to deal with all potential problems faced by the forwarders.

5. Conclusions

In order to develop a suitable algorithm as the core module of the decision support system for air cargo operation, the airfreight forwarder consolidation problem (AFCP) was considered in this study. The AFCP was first transformed to a well-known set covering problem (SCP) by treating a permissible consolidated shipment as a set. The Lagrangian relaxation, a successful approach for SCP reported in prior researches, was used as the backbone to develop a recursive heuristic algorithm. Since the space of the potential sets in the SCP is huge given the numerous ways to consolidate the collected air cargo items, the solution space is initially generated to contain only a limited number of permissible consolidation combinations. A significant amount of effort in this study has been devoted to the procedures for adjusting the solution space through the iterative solution algorithm.

In the numerical experiment, problems with various problem sizes and degrees of concavity were tested. The results show that the heuristic algorithm generates solutions that are very close to optimality for small- and medium-scale problems, whose optimal solutions can be derived from the MIP model. For large-scale problems, with no optimal solution for comparison, it was tested that the algorithm can terminate within a reasonable time. Although the solution algorithm developed in this study appears to be effective in solving the AFCP and should be useful in terms of operational decision support, there remains considerable room to improve the algorithm. For example, the following issues are the directions for research extension:

- A tight lower bound should be developed to ensure the strength of the solution algorithm for large-scale problems.
- The procedure for solution space adjustment should be further improved as it is critical for generating a good quality solution.
- The method to find the feasible solution from the infeasible solution of the relaxed problem needs to be reconsidered given the simple procedure currently used.
- The whole solution procedure may be refined with regards to parameter setting. For example, the exponential smoothing constant and the number of allowed combinations through iterations can be tuned by more computational tests.
- The efficiency of the heuristic algorithm can be improved by advanced coding techniques, so it can deal with large problems within the acceptable time for real application.

Acknowledgements

The authors gratefully acknowledge the support for this project from the National Science Council of Taiwan, through grant no. 92-2211-E-009-052.

References

Amiri, A., Pirkul, H., 1997. New formulation and relaxation to solve a concave-cost network flow problem. Journal of the Operational Research Society 48 (3), 278–287.

Boeing, 2004. World Air Cargo Forecast 2004–2005, Webpage of Boeing Company. Available from: <http://www.boeing.com/commercial/cargo/WACF_2004-2005.pdf>.

Caprara, A., Fischetti, M., Toth, P., 1999. A heuristic method for the set covering problem. Operations Research 47 (5), 730–743.

Held, M., Karp, R.M., 1970. The traveling salesman problem and minimum spanning trees. Operations Research 18, 1138–1162.

Horst, R., Thoai, N.V., 1998. An integer concave minimization approach for the minimum concave cost capacitated flow problem on networks. OR Spektrum 20 (1), 47–54.

Horst, R., Pardalos, P.M., Thoai, N.V., 2000. Introduction to Global Optimization. Kluwer Academic Publishers, The Netherlands.

Irnich, S., 2000. A multi-depot pickup and delivery problem with a single hub and heterogeneous vehicles. European Journal of Operational Research 122, 310–328.

Jaeger, F., 1976. Consolidation strategy for international air freight forwarders – minimum cost routing problem in a directed multi-commodity network. Transportation Research 10, 347–354.

Jordan, W.C., 1986. Scale economies on multi-commodity distribution networks. Report GMR-5579, General Motors Research Laboratories, Warren, MI, USA.

Larsson, T., Migdalas, A., Ronnqvist, M., 1994. A Lagrangian heuristic for the capacitated concave minimum cost network flow problem. European Journal of Operational Research 78 (1), 16–129.

Neter, J., Wasserman, W., Whitmore, G.A., 1993. Applied Statistics. Allyn and Bacon, Needham Heights, MA, USA.

Taillard, E., 1993. Parallel iterative search methods for vehicle routing problems. Networks 23, 661–673.

Ward, J.A., 1999. Minimum-aggregate-concave-cost multi-commodity flows in strong-series-parallel networks. Mathematics of Operations Research 24 (1), 106–129.

Wu, S., 2003. The study of the decision support system for air freight forwarders. MS thesis, National Chiao Tung University, Taiwan.

Xue, J., Lai, K.K., 1997. A study on cargo forwarding decisions. Computers and Industrial Engineering 33, 63–66.

Yan, S., Luo, S., 1998. A tabu-search based algorithm for concave cost transportation network problems. Journal of the Chinese Institute of Engineers 21 (3), 327–335.