AIR CARGO LOADING MANAGEMENT SYSTEMS FOR LOGISTICS FORWARDERS

BY

HAI THI HONG HA

**A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF MASTER OF
ENGINEERING (LOGISTICS AND SUPPLY CHAIN SYSTEMS
ENGINEERING)
SIRINDHORN INTERNATIONAL INSTITUTE OF TECHNOLOGY
THAMMASAT UNIVERSITY
ACADEMIC YEAR 2015**

AIR CARGO LOADING MANAGEMENT SYSTEMS FOR LOGISTICS FORWARDERS

BY

HAI THI HONG HA

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF MASTER OF
ENGINEERING (LOGISTICS AND SUPPLY CHAIN SYSTEMS
ENGINEERING)
SIRINDHORN INTERNATIONAL INSTITUTE OF TECHNOLOGY
THAMMASAT UNIVERSITY
ACADEMIC YEAR 2015

AIR CARGO LOADING MANAGEMENT SYSTEM FOR LOGISTICS FORWARDERS

A Thesis Presented

By

HAI THI HONG HA

Submitted to
Sirindhorn International Institute of Technology
Thammasat University
In partial fulfillment of the requirements for the degree of
MASTER OF ENGINEERING (LOGISTICS AND SUPPLY CHAIN SYSTEMS ENGINEERING)

Approved as to style and content by

Advisor and Chairperson of Thesis Committee ______________________
(Asst. Prof. Dr. Narameth Nananukul)

Committee Member and
Chairperson of Examination Committee ______________________
(Dr. Sun Olapiriyakul)

Committee Member ______________________
(Assoc. Prof. Dr. Parthana Parthanadee)

MAY 2016

Acknowledgements

I would like to express my deepest gratitude to my advisor, Assist. Prof. Dr. Narameth Nananukul for his tremendous enthusiasm, patience, and continuous support for my study and research. Without these, my thesis would not have been possible. I truly admire him as an excellent teacher, supportive advisor and distinguished researcher in operation research.

I am thankful to the rest of my thesis committee: Assoc. Prof. Dr. Parthana Parthanadee and Dr. Sun Olapiriyakul for their questions and motivations along my research.

I owe my gratitude to Sirindhorn International Institute of Technology for my EFS scholarship. Without their trust, I would not have had a chance to experience study and life in Thailand for two years.

I am indebted to friends in Thailand. They deliver their warm welcome and treat me so nicely. It is an honor for me to know and make friends with them.

Very special thanks go out to my best friend, Christoph Walter Senn, who received my arrival in Thailand and showed me the beauty of this country. He also inspired and motivated me along my study and research at SIIT.

Finally, I would like to acknowledge to my family who always support and believe in my decisions.

Abstract

AIR CARGO LOADING MANAGEMENT SYSTEMS FOR LOGISTICS FORWARDERS

by

HAI THI HONG HA

Bachelor in Economics, Foreign Trade University, 2013
Master of Engineering (Logistics and Supply Chain Systems Engineering), Sirindhorn International Institute of Technology, Thammasat University, 2015

The research aims to solve operation problems of an air cargo forwarder that has to manage allotment contracts with several airlines for their single-leg air shipments. An example of a typical international freight forwarder is used to illustrate the air cargo loading management problem. The forwarder makes the bookings in terms of Unit Load Devices (ULDs). In this research, two models were proposed in order to implement air cargo loading management systems for the forwarder; the first is air cargo 3D packing model, and the second model is air shipment assignment model. Air cargo 3D packing model addresses loading shipments of multiple-sizes and released dates into ULDs in order to minimize unused space of ULDs. The model considers overlapping problem, practical position of cartons, cargo priority, weight limitation and other commitments stated by the forwarder. Air shipment assignment model aims to assign shipments among contracted airlines with transportation cost formed as piecewise function. Computational results are provided to demonstrate the performance and limitation of both models. To be able to handle large number of shipments, the relaxed approach for solving the loading model was also proposed. The research distinguishes well from previous studies for combination of two models under a set of constraints not only to assure committed service level to their customers but also to improve cargo operation performance at the forwarder.

Keywords: 3D packing, air cargo loading management, air cargo scheduling, air cargo revenue management.

Table of Contents

List of Figures

List of Tables

Tables **Page**

Chapter 1
Introduction

1.1 Introduction

Air cargo industry has been playing an increasingly important role in transporting high value products with short lead-time. According to a report by Boeing, world air cargo revenue is expected to grow at an average annual rate of 4.7% for the next 20 years from 207.8 billion revenue ton-kilometers (RTK) logged in 2013 (Boeing, 2014). Airlines’ cargo capacity is allocated to intermediates called forwarders, directional customers, and local station managers. Directional customers, who are large clients, usually receive a very favorable rate, and account for a significant portion of the total revenue. Local station managers are essentially freight forwarders owned by the carrier. However, they only focus on the amount of space on flights departing from their stations. Most of cargo airlines’ business is based on several key forwarders who buy capacity from airlines then resell it to their customers. More than 60% of domestic and 90% of international air cargo capacity is sold to forwarders (K. Amaruchkul, W. L. Cooper, & Gupta, 2011). Forwarders buy cargo space on a contract basis or on a request-reply basis (B. Slager & Kapteijns, 2003; J.S. Billings, A.G. Diener, & B.B. Yuen, 2003), both of which account for revenue management in air cargo. In the former, allotment contracts signed between airlines and forwarders allow them to reserve amount of capacity for a predetermined period (six months or a year) on a specific flight or specific time such as weekends or weekdays (B. Slager & Kapteijns, 2003). Allotment contracts (or mid-term allocation processes) are only allocated to big and regular customers that generate a stable revenue stream to the carrier.

The contracts hedge airlines from capacity utilization risk in absent of sales agreements (R. Hellermann, 2006). In return for the risk shift, air cargo carriers charge lower freight on contract capacity to the forwarders. An Airline can accept allotment requests from many forwarders with different policies. Simultaneously, a busy shipper or forwarder probably signs contracts with several airlines with different prices and schedules in order to secure a certain service level committed to their customers. Unlike request-reply basis, on which shippers request capacity only when they are (almost) certain about their shipments, allotment contracts are based on the forecast of regular shipments. Therefore, practically a shipper might deliver less or more cargo than its allotment. So far many literatures have studied air cargo revenue management and allotment options between a single airline – forwarder or an airline with several forwarders. This research focuses on a typical forwarder that has to manage allotment contracts with several carriers for their single-leg air shipments. The research distinguishes well from previous studies for combination of two models under a set of time constraints to not only assure committed service level to their customers but also improve cargo operation performance at the forwarder.

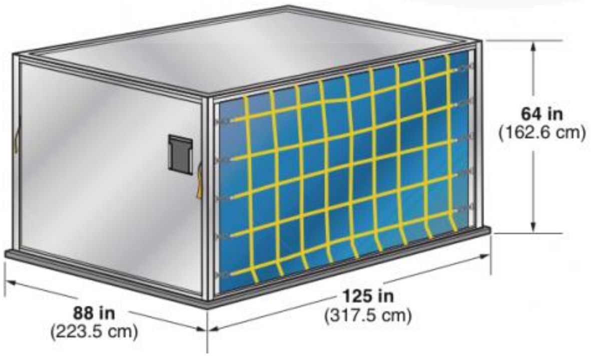
1.2 Statement of the problem

Unlike in spot market, large forwarders always carry a certain amount of capacity in hands according to their contracts with airlines. In general, because of demand uncertainty, forwarders might make under capacity booking with airlines for some weekdays. As a result, this could incur overbooking in other days, which leads to ineffectiveness in operations of both carriers and forwarders. Bookings based on allotment are usually uncertain until departure time; therefore, in many cases, carriers refuse advance bookings from other customers earlier, but end up with much empty space at the time planes take off. Although allotment contracts provide favorable freight rate to forwarders, it is important for forwarder and airlines to avoid wasting space from allotment in order to maximize the efficiency of the operation.

In this research work, two models were proposed to perform daily air cargo loading management. The first model, air cargo 3D packing model, chooses cargoes to be loaded into ULDs in order to maximum volume utilization specified by the allotment contracts. The second model, air shipment assignment model assigns each ULD to certain airline with the objective of minimizing transportation cost while satisfying delivery constraints. The proposed models will be solved sequentially to assure committed service level to the customers and improve cargo operation performance at the forwarder.

An example from a typical international freight forwarder is used to illustrate the air cargo load management. Assume that a freight forwarder has daily shipments that need airfreighting from Hanoi (HAN), Vietnam to Charles de Gaulle (CDG), France. The forwarder has allotment contracts with several airlines, which are Korean Airlines, Vietnam Airline, Singapore Airlines, Qatar Airway. The forwarder usually makes their bookings in terms of Unit Load Devices (ULDs). A ULD is an assembly of components consisting of a container or of a pallet covered with a net, whose purpose is to provide standardized size units for individual pieces of baggage or cargo, and to allow for rapid loading and unloading (S. Limbourg, M. Schyns, & Laporte, 2012). ULDs have some special shapes and measurements which are subject to the operation conditions of the individual airline company. Common ULDs are illustrated in Figure.1.1.

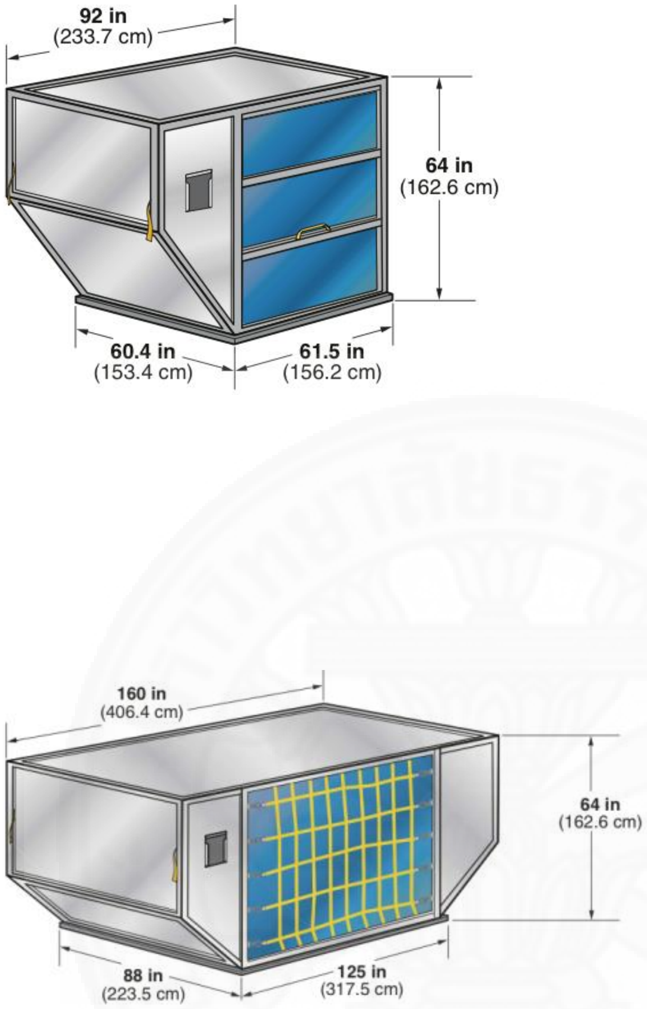
The typical operation at the forwarder can be described as follow: Every Friday, the forwarder receives production plan from customers for next week and makes bookings with contracted airlines for along week. Their major decision is to pack cargoes into ULDs and then assign those ULDs to flights. All the requirements, related to airfreight time, transit time and shipping constraints from customers, must be taken into account in the decision. This research work contributes two aspects of cargo management by building two models. The first model considers loading multiple sized shipments into ULDs, which is subject to constraints from dimension of an ULD stated by the forwarder and delivery time commitment of the shipments. The second model assigns ULDs to airlines based on airline schedules and ready dates of ULDs.



Common Design: LD-9

IATA code: AAP enclosed pallet on P1P.

Maximum gross weight: Lower hold: 4624 kg, Main desk: 6000 kg



Common Design: LD-3

IATA code: AKE contoured container,

Also known as: AKE, AVA, AVB, AVK, DVA, DVE, DVP, XKS, XKG, and fork -able AKN, AVN, DKN, DVN, XKN.

Maximum gross weight: 1588 kg

Common Design: LD-6

IATA code: ALF contoured container,

Also known as: AWA, AWF, and fork -able AWC.

Maximum gross weight: 3175 kg

Figure 1.1 Different shapes of ULDs

1.2.1 Air Cargo 3D packing problem

One of the most functions of the forwarder is to pack cartons of shipments with different sizes and released dates into ULDs. A three dimensional packing model was formulated and tested with different data sets. The model considers overlapping problem, practical position of cartons, cargo priority, weight limitation and other commitment stated by the forwarder. The model provides decision on cargos' position as well as cargos' orientation in each ULD with the objective of minimizing unused space.

1.2.2 Air shipment assignment model for forwarders

The objective of the model is to assign ULDs to airlines with minimum transportation cost. Cost structure includes fixed cost to rent ULDs stated by airlines and the variable cost which is subject to actual chargable weight of ULDs. Cost function is represented by piecewise function which is calculated as below:

Assumed that Airline A has a policy as follow:

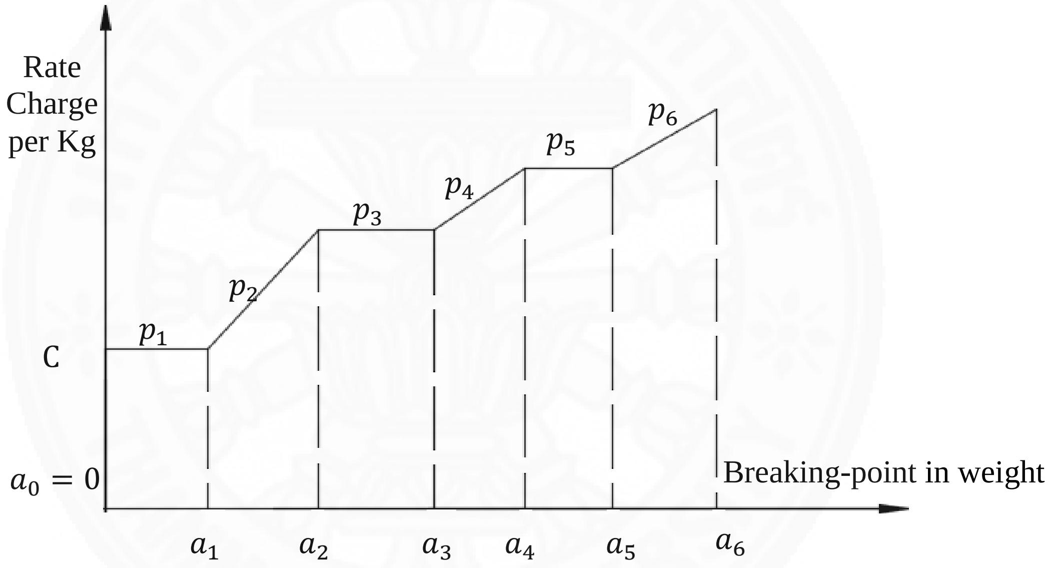


Figure 1. 2 Piecewise function for air cargo variables cost

C: Cost for renting ULD from the airline

w_i: Chargeable weight of ULD i

a_1, .. a_6 : Breaking point in weight stated by airlines

p_1: Price in segment $(a_0, a_1]$

In case ULD i is assigned to airline A. F represents for cost function is as bellow:

$$F= \begin{cases} \mathrm{C} & if\ w_i \in (a_0, a_1] \\ \mathrm{C} +\ p_2.(w_i - a_1) & if\ \ w_i \in (a_1, a_2] \\ \mathrm{C} +\ p_2.(a_2 - a_1) & if\ w_i \in (a_2, a_3] \\ \mathrm{C} +\ p_2.(a_2 - a_1) +\ p_4.(w_i - a_3) & if\ \ w_i \in (a_3, a_4] \\ C +\ p_2.(a_2 - a_1) +\ p_4.(a_4 - a_3) & if\ \ w_i \in (a_4, a_5] \\ \mathrm{C} +\ p_2.(a_2 - a_1) +\ p_4.(a_4 - a_3) +\ p_6.(w_i - a_6) & if\ \ w_i \in (a_4, a_5] \end{cases}$$

Contracts are fixed for a time period; however, to support the forwarder’s handling cargo, airlines allow ad-hoc booking which is beyond allotment contract and charged with spot market prices. Ad-hoc bookings give opportunity for forwarder to access available extra capacity to load their cargos. The prices are subject to market demand at the time of booking, therefore the prices could be higher or lower than contracted prices. The model generates solution that guarantees commitments between forwarder and its customers.

1.3 Thesis Outline

The remaining parts of thesis are organized as follow:

Chapter 2: Literature review. In this chapter, previous researches relating to 3D packing, and air shipment assignment problems, along with popular approaching methodologies are reviewed.

Chapter 3: Air cargo 3D packing model. This section presents 3D packing model mathematically as well as computational results. In addition, the relaxed model is proposed in order to solve the problem within reasonable time.

Chapter 4: Air shipment assignment model. Similar to Chapter 3, air shipment assignment model is presented mathematically, and tested with different data sets.

Chapter 5: Conclusion. This chapter provides some comments and suggestion for logistics forwarders in dealing with allotment contracts. Also, further research is discussed in this section.

Chapter 2
Literature Review

2.1 Air cargo 3D packing problem

Packing problem can be traced back to one-dimensional bin packing or partitioning problem, in which a set of n associated sizes must be divided into the minimum number of subsets so that the sum of sizes in each subset does not exceed a given capacity (Baltacioglu, 2001). Later on, many researchers addressed two-dimensions packing problem with extract approaches and heuristics approaches. The problem is to allocate a set of n heterogeneous rectangular items, defined by width and height, to a minimum number of identical rectangular bins, with the edges of items paralleling to those of the bins but without overlapping (A. Lodi, S. Martello, & Vigo, 2002). Examples of extract approaches for two dimensional packing problem can be found in studies by P.C. Gilmore and Gomory (1965); J. Herz (1972); J. Beasley (1985) and E. Hadjiconstantinou and Christofides (1995). Efficient heuristics approaches were presented in a paper by N. Christofides and C. Whitlock (1977).

Recently, three dimensional packing problems have drawn more attention of researchers. Among huge amount of research on packing problems, to the best of our knowledge, there is a little research with mathematical models which can be solved by standard software package. R. Tsai, E. Malstrom, and Kuo (1993) were the first researcher to provide a three dimensional packing model to load boxes of different sizes into a pallet without overlapping. In 1995, Chen and his colleagues proposed a mixed integer linear programming model to solve 3D container loading problem. The problem was raised to pack a set of non-uniform cartons into unequal-sized containers (Multi-Container Loading Problem-MCLP) with carton orientation and overlapping constraints (C.S. Chen, S.M. Lee, & Q.S. Shen, 1995). Their model took care of non-overlapping (to prevent overlapping between boxes); boxes orientation (to allow boxes to rotate in three dimensions) and container constraint (to recognize certain container is used or not). Although the mathematical model guarantees to lead to optimal total unused space of container(s), a large number of constraints and variables makes the model computationally unfeasible for real-life complex problems. M. Padberg (2000) also introduced a mixed integer model for Knapsack Loading Problem (KLP), in which the objective is to pack maximum number of multiple sized boxes into only one type of container. He estimated the model could test up to 10-20 boxes in a reasonable computing time by standard branch and bound algorithm; however, numerical experiment results were not provided. Another optimal model for MCLP was proposed by J. Westerlund, L. Papageorgiou, and Westerlund (2005). They built the model as an extension of research work on Process Plant Layout (PPL) by L.G. Papageorgiou and G.E. Rotstein (1998) and D.I. Patsiatzis and L.G. Papageorgiou (2002) and Facility Layout (FL) problems by I. Castillo and Westerlund (2005). The proposed model includes additional constraints to support computational effort in cutting down CPU time. Those constraints are non-over-packing constraints and symmetry-breaking constraints. L. Junqueira, R. Morabito, and Yamashita (2012) introduced a mixed integer linear programming model (MIP) for KLP considering constraints of multi-dropping and the vertical stability of the cargo. They solved the problems of packing boxes with different customers (destinations) into containers. In order to avoid additional handling, boxes to be delivered to the same customer should be placed close to each other. A. Bortfeldt and G. Waescher (2013) presented a very detailed classification of packing problems, in concern with three dimensions. More recently C. Paquay, Schyns, and Limbourg (2014) discussed a linear model to pack a set of strongly heterogeneous boxes into ULDs with various shapes for air cargo loading. Many practical constraints in air cargo industry are mathematically formulated in their paper. Examples of the constraints are: Geometric constraints (the small boxes should be stacked above the bigger ones and boxes cannot be overlapped); special shapes of ULDs in air cargo industry, Vertical stability (in airlines industry, when container is not being moved, the constraints prevent items from falling down and withstanding the gravity force). Fragility (some boxes are fragile thus they cannot support other boxes on top of their faces; Weight distribution (or load balance constraints, it requires that the weight of the cargo must be spread evenly across the container floor so that the cargo will not shift while the container is moving). More definition and review of constraints used so far in container loading problem especially in air cargo industry can be found in the paper by A. Bortfeldt and G. Waescher (2013).

2.2 Air Shipment assignment model

How forwarders assign shipments among airlines, in earlier literature, is a sub-problem of air freight forwarder consolidation problem (AFCP).

Xue and Lai (1997) handled container rental and cargo shipment problem of an air freight forwarder in Hongkong. They formulated a mixed integer programming model to minimize total of fixed container rental cost and variable cost which is a piece-wise linear function

Huang and Chi (2007) suggested transforming AFCP into set cover problem (SCP) which can be approached with Lagrangian Relaxation. Based on this, they developed a recursive heuristics algorithm in order to solve large-scale problems within a reasonable time.

Y. Li, Tao, and Wang (2009) studied daily operation of a forwarder, that paid capacity reservation from airlines every morning based on their forecasting and made cargo loading plan in the afternoon. An integer programming model was proposed to load cargo into a given number of ULDs with different fixed reservation fees, pivot weight and unit cost. The objective is to minimize total cost and implement large- scale neighborhood search heuristic in practical time. However the research ignored dimensions of cargo when assigning cargo to containers.

Leung, Van Hui, Wang, and Chen (2009) divided forwarding process into multiple jobs and addressed problem of integration and consolidation of air cargo shipments during any phase of the process. A linear 0-1 programming was formulated to get the best job-shipment assignment with minimal operation cost. They presented both a branch and bound (B&B), and a heuristics algorithm to solve the problem.

Based on the model by Leung et al. (2009), Wong, Leung, and Hui (2009) extend shipment integration and consolidation problem with constraints from delivery time, capacity limitation, and target cost. They provided mixed 0-1 linear programming formulations to describe a forwarder’s shipment planning problem and proposed a customized tabu search algorithm to solve it.

Z. Li, Bookbinder, and Elhedhli (2012) introduced mixed integer programming model to optimize shipment planning problems of an airfreight forwarder in a network with multiple origins and destinations. Their model use piecewise-linear cargo rates and is subject to volume and weight constraints, flight departure/arrival time, shipment delivery time, and bumping clause (the possibility of over-declaration to the next weight range for more favorable rate). To solve large size problems, they proposed two methodologies, which are Lagrangian relaxation to decompose problem into a set of knapsack problems and a set of network flow problems, and local branching heuristic.

Chapter 3
Air cargo 3D packing model

3.1 Mathematical model for air cargo 3D packing

In this section, an air cargo packing model is introduced and tested with a set of sample data. The data set was used by the forwarder to make booking with contracted airlines. Most ULDs are designed to fit into cargo compartment of individual airline, the dimension of ULDs in this research is fixed by the forwarder as 317.5 cm (L) x 223.5 cm (W) x 162.6 cm (H). The ULD type is LD-9 and has IATA code APA. In general, it fits to lower compartment of aircraft.

In this research work, an existing packing model was enhanced by adding constraints incurred from the allotment contracts and commitment between forwarder and their customers. The proposed model takes into account loading priority. Maximal total weight of cartons in each ULD is considered. Additionally, typical commitment by the forwarder to their customers which makes sure that cartons must be air freighted within 2 days after being released is also consider in the model. Precondition to satisfy the previous requirement is that each ULD can carry cartons released at most more than 2 days apart from each other. The parameters and variables are set as follow:

Parameters:

O: Set of ULDs available in a week

N: Set of cartons available in a week

M: An arbitrarily large number.

p_i, q_i, r_i: Length, width, and height of carton i

we_i : weight of carton i

v_i : Released date of carton i

pr_i : Co-efficient representing priority level of carton i

L, H, W: Length, height, width of ULD

Variables:

n_i : A binary variable which is equal to 1 if the ULD j is used; otherwise it is equal to 0.

S_{ij} : Binary variable which is equal to 1 if carton number i is placed in ULD j.

m_j : Weight of ULD j after packing

x_i, y_i, z_i : Continuous variables (for location) indicating the coordinates of the front-left bottom (FLB) corner of carton i placed in the ULD.

l_{xi}, l_{yi}, l_{zi} : Binary variables which are equal to 1 if the length of carton i packed is parallel to the X, Y or Z, otherwise they are equal to 0.

w_{xi}, w_{yi}, w_{zi} : Binary variables which are equal to 1 if the width of carton i packed is parallel to the X, Y or Z, otherwise they are equal to 0.

h_{xi}, h_{yi}, h_{zi} : Binary variables which are equal to 1 if the height of carton i packed is parallel to the X, Y or Z, otherwise they are equal to 0.

$a_{ik}, b_{ik}, c_{ik}, d_{ik}, e_{ik}, f_{ik}$: Binary variables which are equal to 1 if box i is on the left, right, in front, behind, bellow, upper side of carton k, respectively, otherwise, they are equal to 0.

$TotalVol_j$: Total volume of ULD j after packing

$UsedVol_j$: Used volume of ULD j after packing

Table 3.1 Variables and parameters

$x_i = 0$ $y_i = 0$ $z_i = 0$	$h_{xi} = 1$ $w_{yi} = 1$ $l_{zi} = 1$	$x_k = 149$ $y_k = 99$ $z_k = 0$	$w_{xk} = 1$ $h_{yk} = 1$ $l_{zk} = 1$	$a_{ik} = 1$ $b_{ik} = 0$ $c_{ik} = 1$	$d_{ik} = 0$ $e_{ik} = 0$ $f_{ik} = 0$

Table 3.1 provides an example of sets of variables and parameters of two cargos inside the ULD. A visual image of cargos inside ULD, as can be seen in Figure 3.1, was drawn from data inTable 3.1.

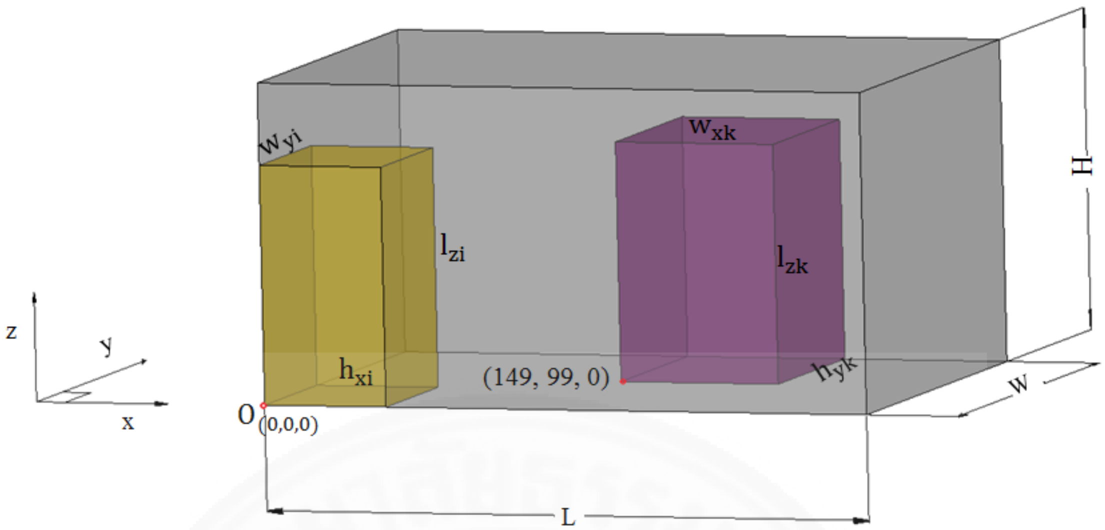


Figure 3. 1 Variable definitions

$$\text{Objective function: Minimize } \sum_{j=1}^{O} L.W.H.n_j - \sum_{i=1}^{N} p_i.q_i.r_i.pr_i \qquad (1)$$

The first part of objective function is the total volume of ULDs used. The second part, which is the adjusted total volume of cartons inside ULDs, takes cargo priority into account. The objective function is to minimize number of ULDs used and unused space of ULDs.

Constraints:

$$\sum_{j=1}^{O} \mathrm{S_{ij}} = 1 \quad \forall i \in N \qquad (2)$$

$$\sum_{i}^{N} we_i.S_{ij} \leq 4500 \quad \forall j \in O \qquad (3)$$

$$|v_k - v_i| \leq 2 + M.(1 - S_{ij}) + M.(1 - S_{kj}) \quad \forall \mathrm{j} \in \mathrm{O}, \text{and i, k} \in \mathrm{N} : \mathrm{k} > \mathrm{i} \qquad (4)$$

$$m_j = \sum_{i}^{N} we_i.S_{ij} \quad \forall j \in N \qquad (5)$$

$$\sum_{i}^{N} S_{ij} \leq M.n_j \quad \forall j \in O \qquad (6)$$

$$TotalVol_j = L.W.H.n_j \quad \forall j \in O \qquad (7)$$

$$UsedVol_j = \sum_{i}^{N} p_i.q_i.r_i.S_{ij} \quad \forall j \in O \qquad (8)$$

Constraint (2) ensures that each carton will be placed in exactly one ULD. Constraint (3) sets upper bounds for weight of containers which is subject to allotment contracts. Constraint (4) ensures that in every pallet, there are no 2 cartons released in more than 2 days later than each other. Constraint (5) ensures that the weight of any

ULD is equal to sum of weight of carton packed inside it. Constraint (6) describes if any carton is assigned to an ULD, the ULD is considered used. Constraints (7) and (8) describe how to calculate total volume and used volume of ULD after packing.

Non- overlapping constraints:

$\forall i, k \in N / k > i :$

$\forall i, k \in N / k > i :$

$$x_i + p_i.l_{xi} + q_i.w_{xi} + r_i.h_{xi} \leq x_k + (1 - a_{ik}).M \tag{9}$$

$$x_k + p_k.l_{xk} + q_k.l_{xk} + r_k.h_{xk} \leq x_i + (1 - b_{ik}).M \tag{10}$$

$$y_i + p_i.l_{yi} + q_i.w_{yi} + r_i.h_{yi} \leq y_k + \left(1 - c_{ik}\right).M \tag{11}$$

$$y_k + p_k.l_{yk} + q_k.w_{yk} + r_k.h_{yk} \leq y_i + \left(1 - d_{ik}\right).M \tag{12}$$

$$z_i + p_i.l_{zi} + q_i.w_{zi} + r_i.h_{zi} \leq z_k + \left(1 - e_{ik}\right).M \tag{13}$$

$$z_k + p_k.l_{zk} + q_k.w_{zk} + r_k.h_{zk} \leq z_k + \left(1 - f_{ik}\right).M \tag{14}$$

Constraints (9)-(14) describe that the cartons can rotate orthogonally in the container and there is no overlapping between any two cartons.

$\forall i, k \in N :$

$$z_k \leq e_{ik}.H \tag{15}$$

$$z_i \leq f_{ik}.H \tag{16}$$

Constraints (15) and (16) force the coordinate of the front-left bottom (FLB) corner of carton along z equal to 0 when there is no another carton bellow it.

$$a_{ik} + b_{ik} + c_{ik} + d_{ik} + e_{ik} + f_{ik} \geq S_{ij} + S_{kj} - 1 \quad \forall i, k \in N : k > i \tag{17}$$

$\forall j \in O,\ i \in N :$

$$x_i + p_i.l_{xi} + q_i.w_{xi} + r_i.h_{xi} \leq L + \left(1 - S_{ij}\right).M \tag{18}$$

$$y_i + p_i.l_{yi} + q_i.w_{yi} + r_i.h_{yi} \leq W + \left(1 - S_{ij}\right).M \tag{19}$$

$$z_i + p_i.l_{zi} + q_i.w_{zi} + r_i.h_{zi} \leq H + \left(1 - S_{ij}\right).M \tag{20}$$

Constraint (17) guarantees relative position of each pair of cartons, which also checks for overlapping. Constraint (18) - (20) ensures that cartons do not exceed their ULD’s dimension.

Since the length (or the width, or height) of cargos must be parallel to one of axes X or Y or Z:

$$l_{xi} + l_{yi} + l_{zi} = 1 \tag{21}$$

$$w_{xi} + w_{yi} + w_{zi} = 1 \tag{22}$$

$$h_{xi} + h_{yi} + h_{zi} = 1 \tag{23}$$

Also, the length or width or height (one of three dimension of the carton) must be parallel to X (or Y, or Z):

$$l_{xi} + w_{xi} + h_{xi} = 1 \tag{24}$$

$$l_{yi} + w_{yi} + h_{yi} = 1 \tag{25}$$

$$l_{zi} + w_{zi} + h_{zi} = 1 \tag{26}$$

3.2 Computational results

This section aims to test the limit of using CPLEX to solve air cargo 3D packing problem. Several instances of data set with different numbers of cargoes and ULDs were collected from practical cargo shipping business. Computer used for testing has CPU 3.30 Hz, 8.00GB RAM, and 64-bit operating system and, is equipped with IBM ILOG CPLEX Optimization Studio version 12.3. Data set for testing is from subsets of a full data file provided by the forwarder with 315 cargoes to be packed in week 4 of the year 2014 (20/Jan/2014 to 26/Jan/2014). The Table 3.2 shows a sample data set used for testing. The first column is numbers of Delivery Order (D/O) which specifies the shipments to be air freighted. Those shipments are packed into cartons by the forwarder’s customer. Those cartons have specifications about weight (kg), height (cm), width (cm), length (cm) and model as shown in the Table 3.2. All the shipments are transported from Hanoi international airport, Vietnam to Charles de Gaulle (CDG), France. SOP stands for standard operation procedure. In this study, SOP states the specified week in which shipment must be shipped. The sample data set are from 20/Jan/2014 to 26/Jan/2014, for planning in week 04 of the year (2014); therefore, shipments with SOP W04 must be taken into priority over W05 and W06 to be shipped within the week (Week04). In order to represent the priority requirement of the model, SOPs are weighted according to the priority levels (for W04, W05, and W06, the weights equal 3, 2, and 1 respectively). Forecast column informs date of cargo readiness. An example of the result from air cargo 3D packing model is shown in Table 3.3.

Table 3.2 A sample data set for testing model

DO	Weight	Height	Width	Length	Volume	Model	Port	SOP	Forecast	Priority
7683321	297	63	85	122	653310	A-T110NDWAXLD	CDG	W04	20-Jan	3
7683323	297	64	83	121	642752	A-T110NDWAXLD	CDG	W04	20-Jan	3
7683326	297	64	84	121	650496	A-T110NDWAXLD	CDG	W04	20-Jan	3
7837405	298	63	85	122	653310	A-T110NDWAXLD	CDG	W04	21-Jan	3
7837406	296	65	84	122	666120	A-T110NDWAXLD	CDG	W04	21-Jan	3
7831955	370	106	83	121	1064558	A-N9005ZKEBZU	CDG	W05	21-Jan	2
7835466	350	97	85	123	1014135	A-N9005ZKEBZU	CDG	W05	21-Jan	2
7943010	356	120	81	99	962280	D-I9195ZKAXLD	CDG	W04	22-Jan	3
7954858	350	115	84	100	966000	D-I9300OKDXLD	CDG	W04	22-Jan	3
7954885	65	52	60	80	249600	D-N8000ZWAXLD	CDG	W05	22-Jan	2
8139534	163	60	68	110	448800	D-I9300OKDXLD	CDG	W04	23-Jan	3
8139570	430	117	84	120	1179360	D-I9506ZWAFOS	CDG	W05	23-Jan	2
7960357	77	55	61	96	322080	A-V7000ZKAXLD	CDG	W04	24-Jan	3
8140795	322	108	108	123	1434672	D-S5360UWAXLD	CDG	W05	24-Jan	2
8140875	328	100	110	114	1254000	A-N9005WDEXLD	CDG	W04	24-Jan	3

This computational time is around twelve seconds (00:00:11:84) with an optimal solution (-80360608). This model contains 1705 constraints and 1809 variables. The results are summarized in Table 3.3, Table 3.4 and Figure 3.2.

Table 3.3 Cargoes assignment result

No	DO	ULD 1	ULD 2	No	DO	ULD 1	ULD 2
1	7683321	0	1	8	7943010	1	0
2	7683323	0	1	9	7954858	1	0
3	7683326	0	1	10	7954885	1	0
4	7837405	0	1	11	8139534	1	0
5	7837406	0	1	12	8139570	1	0
6	7831955	0	1	13	7960357	1	0
7	7835466	0	1	14	8140795	1	0
				15	8140875	1	0

As can be seen from Table 3.3, fifteen cargos are packed into 2 ULDs with 8 cartons in ULD 1 and 7 cargos in ULD 2.

Table 3.4 Result of coordinates of the front-left bottom (FLB) corner of carton

No	D/O	**x**	**y**	**z**	No	DO	**x**	**y**	**z**
1	7683321	0	83	0	8	7943010	84	0	0
2	7683323	0	0	0	9	7954858	0	123	0
3	7683326	196	0	0	10	7954885	84	171	0
4	7837405	171	84	0	11	8139534	84	99	0
5	7837406	131	0	0	12	8139570	0	0	0
6	7831955	234	84	0	13	7960357	154	99	0
7	7835466	63	84	0	14	8140795	209	0	0
					15	8140875	217	108	0

Information from Table 3.4 addresses position of cargos inside ULDs. Figure 3.2 describes ULD 2 after packing drawn with AutoCad2015. As can be seen from the Figure 3.2, seven cargos were rotated within ULDs to maximize the used space of the ULD.

.

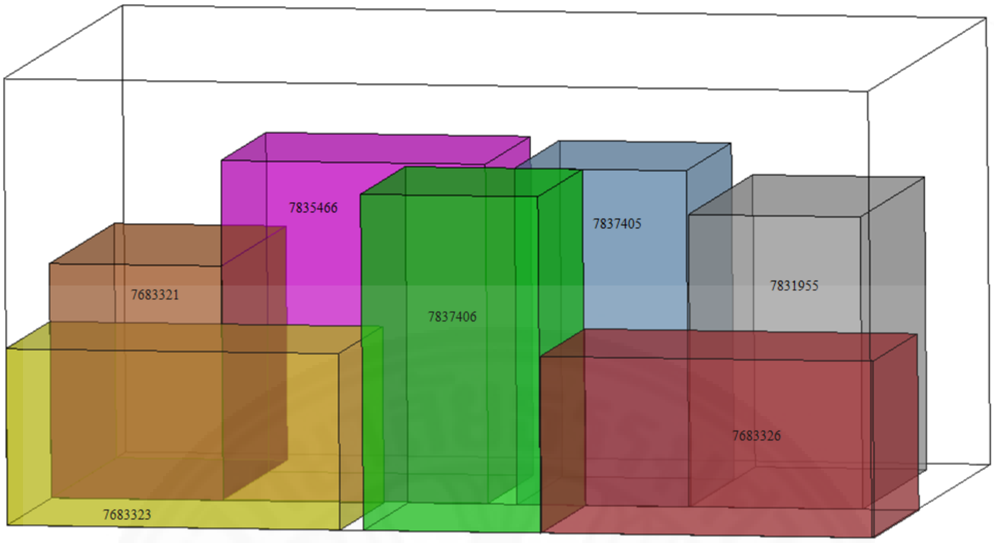


Figure 3.2 ULD 2 after packing

Table 3.5 Cplex test results with different data sets

Data Set#	#carton used	# Constraint	# Variable	Time	Objective	Gap
2ULDs x 15 cartons		1705	1089			
1	2			00:00:19:79	-82155744	0.00%
2	2			00:00:21:90	-87154608	0.00%
3	2			00:00:21:44	-94878564	0.00%
4	2			00:00:21:41	-76254528	0.00%
5	2			00:00:21:49	-82804420	0.00%
3ULDsx 20 cartons		3415	1873			
1	3			02:00:00:00	-359304858	9.56%
2	3			00:28:09:11	-335165112	0.00%
3	3			02:00:00:00	-241707870	14.21%
4	3			02:00:00:00	-374282307	9.18%
5	-			00:59:21:75	Out of memory	
5ULDs x 40 cartons		16585	7021			
1	-			02:00:00:00	Time limit exceeded	
2	-			00:06:10:52	Out of memory	
3	-			00:09:04:80	Out of memory	
4	-			00:04:23:90	Out of memory	
5	-			02:00:00:00	Time Limit Exceeded	

Since different configurations require different amount of computational time for running, different data sets were generated randomly via excel to test the runtime and gap of model Searching time is set at 7200 seconds (2 hours). Table 3.5 shows the computational results after solving the test data with Cplex.

As can be seen from the Table 3.5, a change in configuration results in a significant change in runtimes. In some cases, Cplex fails to generate solution because specified time limit is violated or because of out-of-memory issue. The new approach is based on the decomposition of the original model into a relaxed model where orientation related constraints are removed. The relaxed model is solved with difference configuration until the position of all cargos in ULDs is satisfied.

3.3 Relaxed Model

In this section, a relaxed model, which relaxes the group of orientation constraints from constraints (9) – (12) is proposed. Also, the relaxed model is extended to consider the case where not all cargos can be packed for shipping therefore, choice of packing cargos is based on their priorities. The model is described as follow:

Parameters:

O: Set of ULDs available in a week

N: Set of cartons available in a week

M: An arbitrarily large number

$vol_i = p_i.q_i.r_i$: Volume of carton i

we_i : weight of carton i

v_i : Released date of carton i

pr_i : Co-efficient representing priority level of carton i

L, H, W: Length, height, width of ULD

Fd : Maximum usage percentage of ULD

Variables:

n_i : A binary variable which is equal to 1 if the pallets j is used; otherwise it is equal to 0.

S_{ij} : Binary variable which is equal to 1 if carton number i is placed in pallet j.

m_j : Weight of ULD j after packing

$TotalVol_j$: Total volume of ULD j after packing

$UsedVol_j$: Used volume of ULD j after packing

Objective function: Minimize $\sum_{j=1}^{O} L.W.H.n_j - \sum_{i=1}^{N} p_i.q_i.r_i.$ (1)

Objective function is to minimize unused space of ULDs.

Constraints:

$$\sum_{j=1}^{O} \mathrm{S_{ij}} <= 1 \quad \forall i \in N \tag{2i}$$

$$\sum_{j=1}^{O} \mathrm{S_{i_1 j}} \le \sum_{j=1}^{O} \mathrm{S_{i_2 j}} = 1 \quad \forall i_1, i_2 \in N; pr_{i_2} \ge pr_{i_1} \tag{2ii}$$

$$\sum_{i}^{N} we_i.S_{ij} \le 4500 \;\; \forall j \in O \tag{3}$$

$$|v_k - v_i| \le 2 + M.(1 - S_{ij}) + M.(1 - S_{kj}) \;\; \forall \mathrm{j} \in \mathrm{O, and\ i, k} \in \mathrm{N : k > i} \tag{4}$$

$$m_j = \sum_{i}^{N} we_i.S_{ij} \quad \forall i \in N \tag{5}$$

$$\sum_{i}^{N} S_{ij} \le M.n_j \quad \forall j \in O \tag{6}$$

$$TotalVol_j = L.W.H.n_j \quad \forall j \in O \tag{7}$$

$$UsedVol_j = \sum_{i}^{N} vol_i.S_{ij} \quad \forall j \in O \tag{8}$$

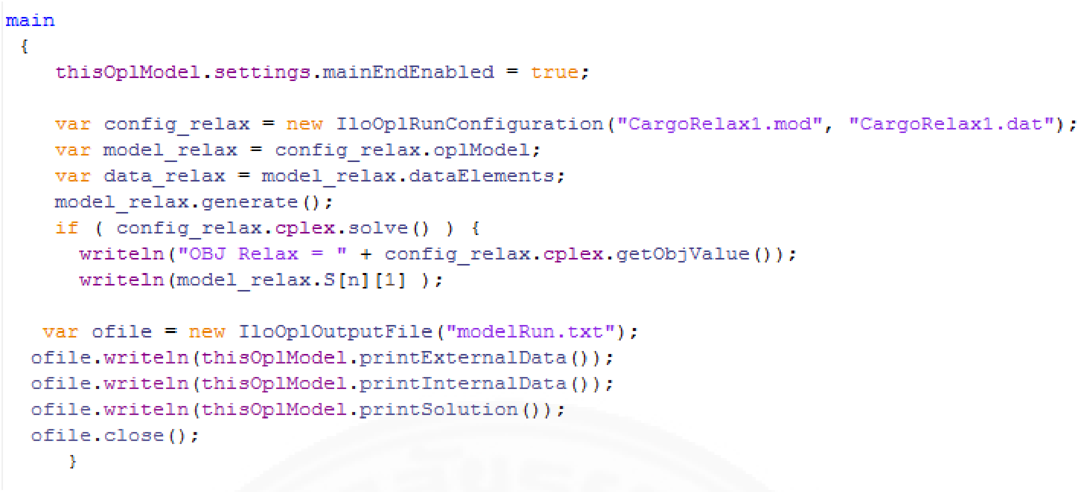
Constraints (2i) and (2ii) make sure that cargos with higher priority co-efficient are taken into priority over others.

Constraints (3) – (8) are similar with those from original model.

$$UsedVol_j \le Fd.TotalVol_j \quad \forall j \in O \tag{9'}$$

Added constraint (9') specifies the percentage of ULDs' volume that can be used.

Results from the relaxed model are processed with 3D packing model with scripts as follow:



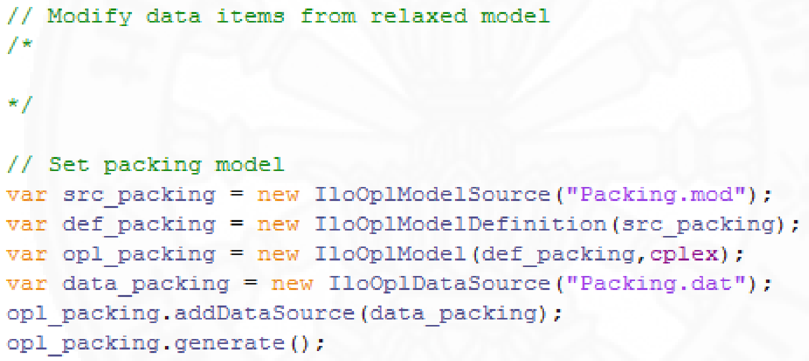
```
main
 {
    thisOplModel.settings.mainEndEnabled = true;

    var config_relax = new IloOplRunConfiguration("CargoRelax1.mod", "CargoRelax1.dat");
    var model_relax = config_relax.oplModel;
    var data_relax = model_relax.dataElements;
    model_relax.generate();
    if ( config_relax.cplex.solve() ) {
      writeln("OBJ Relax = " + config_relax.cplex.getObjValue());
      writeln(model_relax.S[n][1] );

   var ofile = new IloOplOutputFile("modelRun.txt");
  ofile.writeln(thisOplModel.printExternalData());
  ofile.writeln(thisOplModel.printInternalData());
  ofile.writeln(thisOplModel.printSolution());
  ofile.close();
     }
```

Figure 3.3 Script to call relaxed model

Figure 3.3 provides the script to control solving process, which starts with relaxed model first.



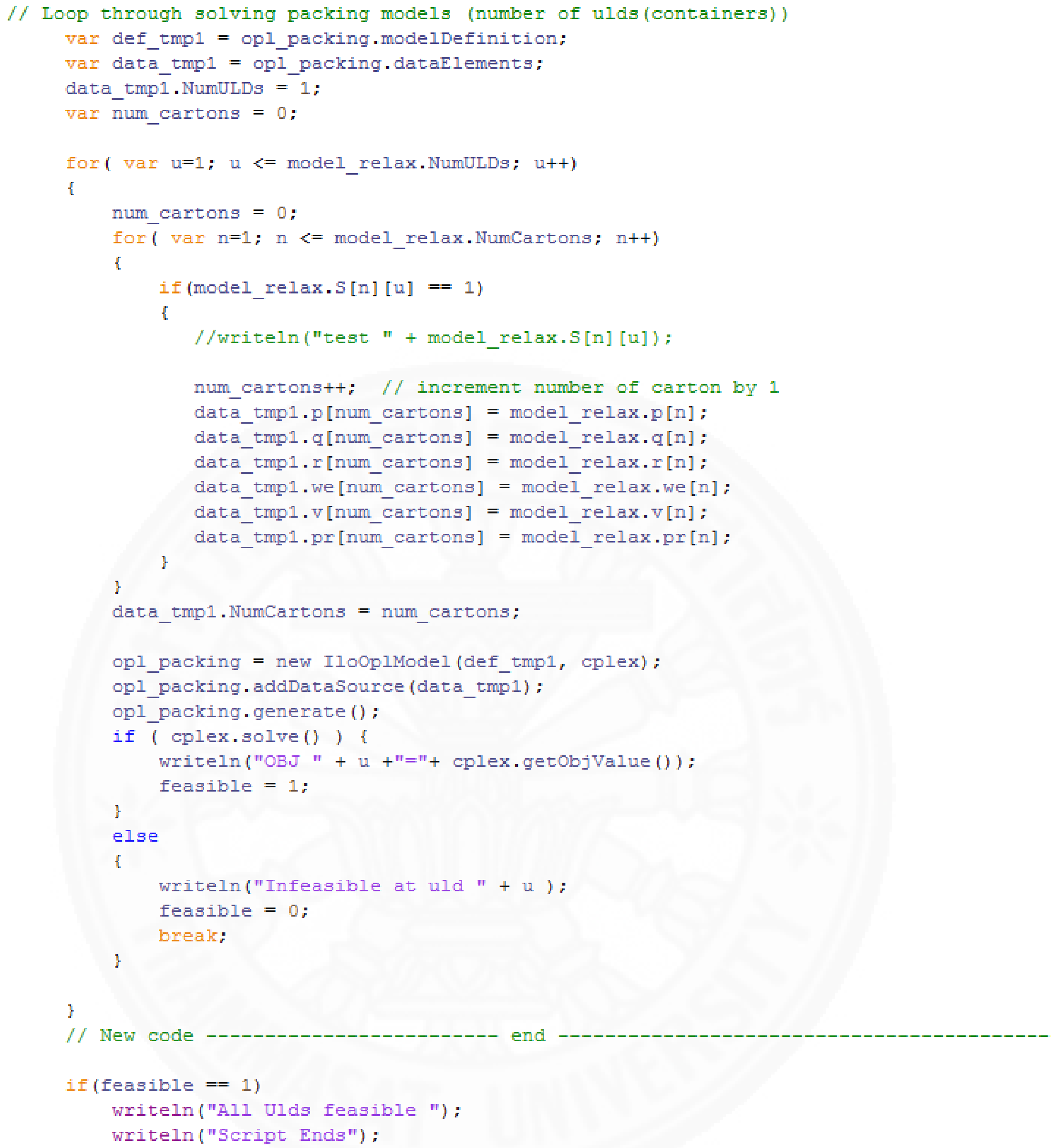
```
// Modify data items from relaxed model
/*

*/

// Set packing model
var src_packing = new IloOplModelSource("Packing.mod");
var def_packing = new IloOplModelDefinition(src_packing);
var opl_packing = new IloOplModel(def_packing,cplex);
var data_packing = new IloOplDataSource("Packing.dat");
opl_packing.addDataSource(data_packing);
opl_packing.generate();
```

Figure 3.4 Script to call packing model

The script in Figure 3.4 integrates packing model into process so that the results from relaxed model can be processed next. The script from Figure 3.5 select cargos assigned to each ULD by relax model to test with orientation constraints of 3D packing model.



```
// Loop through solving packing models (number of ulds(containers))
     var def_tmp1 = opl_packing.modelDefinition;
     var data_tmp1 = opl_packing.dataElements;
     data_tmp1.NumULDs = 1;
     var num_cartons = 0;

     for( var u=1; u <= model_relax.NumULDs; u++)
     {
         num_cartons = 0;
         for( var n=1; n <= model_relax.NumCartons; n++)
         {
             if(model_relax.S[n][u] == 1)
             {
                //writeln("test " + model_relax.S[n][u]);

                num_cartons++;  // increment number of carton by 1
                data_tmp1.p[num_cartons] = model_relax.p[n];
                data_tmp1.q[num_cartons] = model_relax.q[n];
                data_tmp1.r[num_cartons] = model_relax.r[n];
                data_tmp1.we[num_cartons] = model_relax.we[n];
                data_tmp1.v[num_cartons] = model_relax.v[n];
                data_tmp1.pr[num_cartons] = model_relax.pr[n];
             }
         }
         data_tmp1.NumCartons = num_cartons;

         opl_packing = new IloOplModel(def_tmp1, cplex);
         opl_packing.addDataSource(data_tmp1);
         opl_packing.generate();
         if ( cplex.solve() ) {
             writeln("OBJ " + u +"="+ cplex.getObjValue());
             feasible = 1;
         }
         else
         {
             writeln("Infeasible at uld " + u );
             feasible = 0;
             break;
         }

     }
     // New code ------------------------- end -------------------------------------------

     if(feasible == 1)
         writeln("All Ulds feasible ");
         writeln("Script Ends");
```

Figure 3.5 Script for testing packing model for multiple ULDs

Also, script in Figure 3.5 allows to repeat packing ULDs which are assigned by Relaxed model one by one, from the first to the last ULDs.

3.3.1 Bi-section search for maximum usage percentage of ULD

This section suggests bisection search to find the maximum usage percentage of ULD. The bisection method is the simplest way to find roots (solution) of equations. The Bisection (Binary search) Method is based on the Intermediate Value Theorem (IVT). Figure 3.6 describes Bisection search. Suppose a continuous function f, defined on [a, b] is given with $f_{(a)}$ and $f_{(b)}$ of opposite sign ($f_{(a)} \cdot f_{(b)}$<0). By the IVT, there exists a point p ∈ [a, b] for which $f_{(p)}$ = 0. The method calls for a repeated halving (or bisecting) of subintervals of [a, b] and, at each step, locating the half containing p.

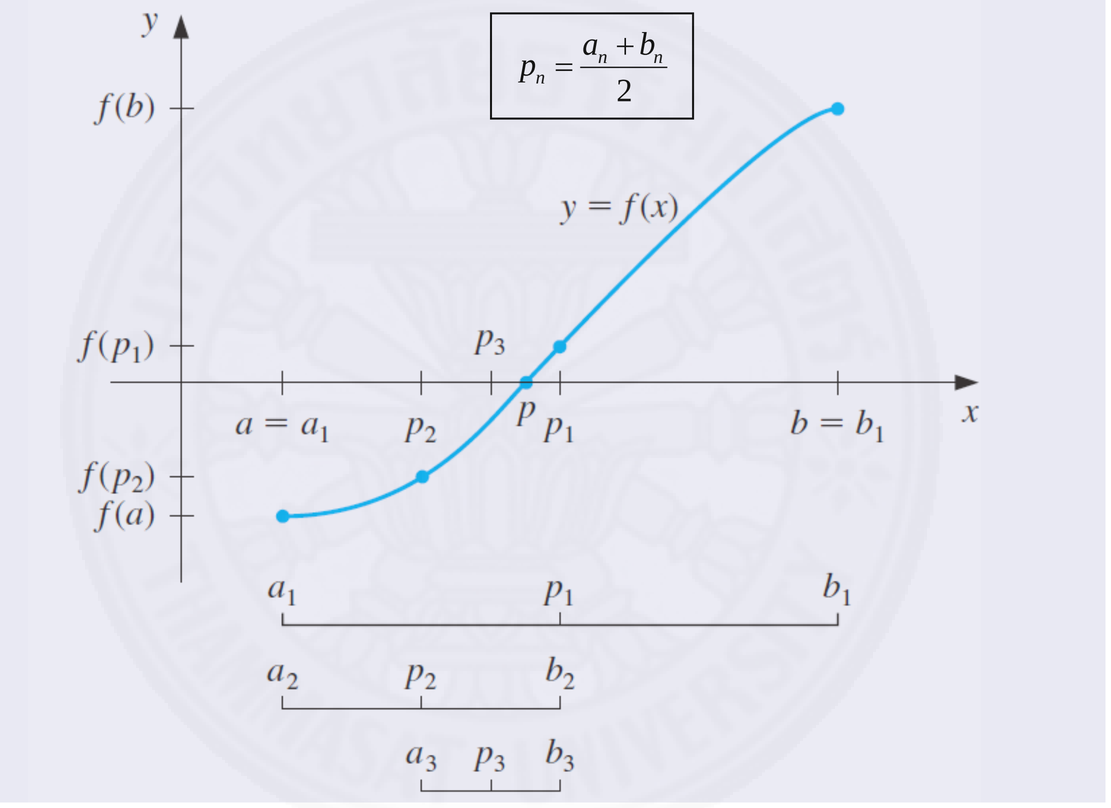


Figure 3. 6 Bisection search

In case where there is no practical clue to set optimal maximum usage percentage, undoubtedly, optimal maximum usage percentage is between 0 and 100%. A data set in Table 3.6 is selected to show how the model works. Similar to Table 3.2, Table 3.6 provides information about weight, 3-dimensional measurements, cargo priory, and forecasted released date. In this table, cargos with SOP W04 have higher priority than others.

Table 3.6 A sample data set for bisection search

No	DO	Weight	Height	Width	Length	Vol	Model	Port	SOP	Forecast	Priority
1	7837395	296	64	85	122	663680	A-T110NDWAXLD	CDG	W04	21	3
2	7837411	296	65	84	122	666120	A-T110NDWAXLD	CDG	W04	22	3
3	7837418	296	65	85	122	674050	A-T110NDWAXLD	CDG	W04	22	3
4	7526102	192	70	84	121	711480	D-I9305OKDXLD	CDG	W04	21	3
5	7829608	254	108	82	120	1062720	D-I9195ZWAXLD	CDG	W04	21	3
6	7943032	387	116	83	111	1068708	D-I9300RWDXLD	CDG	W04	22	3
7	7954944	296	63	84	121	640332	A-T110NDWAXLD	CDG	W04	22	3
8	8139532	421	117	83	120	1165320	D-I9300OKDXLD	CDG	W04	23	3
9	7960313	298	64	84	122	655872	A-T110NDWAXLD	CDG	W04	23	3
10	8195588	294	124	81	121	1215324	A-T110NYKAXLD	CDG	W04	23	3
11	8192222	276	122	82	98	980392	A-P6000ZKAXLD	CDG	W05	23	2
12	8181257	419	116	84	121	1179024	D-I9300MBDXLD	CDG	W04	23	3
13	8301634	376	121	120	116	1684320	D-S5360UWAXLD	CDG	W05	23	2
14	8323558	328	123	80	120	1180800	A-P6000ZKAXLD	CDG	W05	24	2
15	8304662	316	123	83	123	1255707	D-P5210MKAXLD	CDG	W05	24	2
16	8301726	292	121	81	96	940896	A-N9005ZWESPA	CDG	W05	24	2
17	8323561	294	122	83	103	1042978	A-P6000ZKAXLD	CDG	W05	24	2
18	8301640	367	121	120	112	1626240	D-S5360UWAXLD	CDG	W05	24	2
19	8195605	296	124	82	122	1240496	A-T110NYKAXLD	CDG	W04	24	3
20	8625446	176	66	81	70	374220	A-T110NDWAXLD	CDG	W04	25	3

From the data set, we have:

Mean Cargo Volume = $\mu_{vol_i} = \frac{\sum_{1}^{N} vol_i}{N} = 1001434$

ULD Volume = $TotalVol_j = L.W.H = 15337728$

Tolerance = $\varepsilon = \frac{\mu_{vol_i}}{TotalVol_j} = 0.065$

The bisection searching process stops when either criterion is met:

- The objective function remains unchanged. It means the total unused space reaches optimal. Hardly can we reduce a number of ULD.
- After n^{th} iteration, $|p_n - p_{n-1}| - \varepsilon \leq 0$

Table 3. 7 Bisection search results of given data

Iteration No	Min Value (%)	Mid Point (%)	Max Value (%)	Selected Range
1	0	50	100	Upper Range
2	50	75	100	Lower Range
3	50	62.5	75	Upper Range
4	62.5	68.75	75	Lower Range
5	62.5	65.625	68.75	Upper Range

Iteration No	Objective Function		$\mid p_n - p_{n-1} \mid - \varepsilon$	Time
1	25779089	Feasible		00:00:11:66
2		Out of time	0.19	01:00:00:00
3	14327147	Feasible	0.065	00:00:06:13
4		Infeasible	0.0025	00:00:14:79
5	14327147	Feasible	-0.02875	00:00:13:66

Table 3.7 provides information about values of parameter Fd and testing results for each iteration. As can be seen, iteration 5 satisfies$| p_n - p_{n-1} | - \varepsilon \leq 0$. Therefore, acceptable optimal parameter can be set at 65.625%. Bisection search is the simplest way to carry out; however many iterations are probably required.

Table 3.8 Cargo assignment results with Fd= 65.625%

Cargos	ULD 1	ULD 2	ULD 3	ULD 4
1	1	0	0	0
2	1	0	0	0
3	1	0	0	0
4	1	0	0	0
5	1	0	0	0
6	1	0	0	0
7	1	0	0	0
8	1	0	0	0
9	1	0	0	0
10	0	1	0	0
11	0	1	0	0
12	0	1	0	0
13	0	1	0	0
14	0	1	0	0
15	0	1	0	0
16	0	0	1	0
17	0	0	1	0
18	0	0	1	0
19	0	0	1	0
20	0	0	1	0

As can be seen from Table 3.8, given 20 cargos are packed within 3 ULDs, in which the ULD 1 holds the most cargoes, cargos 1 to 9, when Fd is set at 65.625%.

Table 3. 9 Coordinate results for cargo in ULD 1

Cargos	x	y	z
1	0	0	0
2	0	85	0
3	169	108	0
4	247	117	0
5	152	0	0
6	84	112	0
7	0	160	0
8	234	0	0
9	88	0	0

Table 3. 10 Coordinate results for cargo in ULD 2

Cargos	x	y	z
10	193	0	0
11	0	0	0
12	80	123	0
13	201	81	0
14	0	100	0
15	82	0	0

Table 3. 11 Coordinate results for cargo in ULD 3

Cargos	x	y	z
16	0	83	0
17	0	0	0
18	122	82	0
19	129	0	0
20	251	0	0

Table 3.9, Table 3.10, Table 3.11 provide information on position of each cargo inside ULDs.

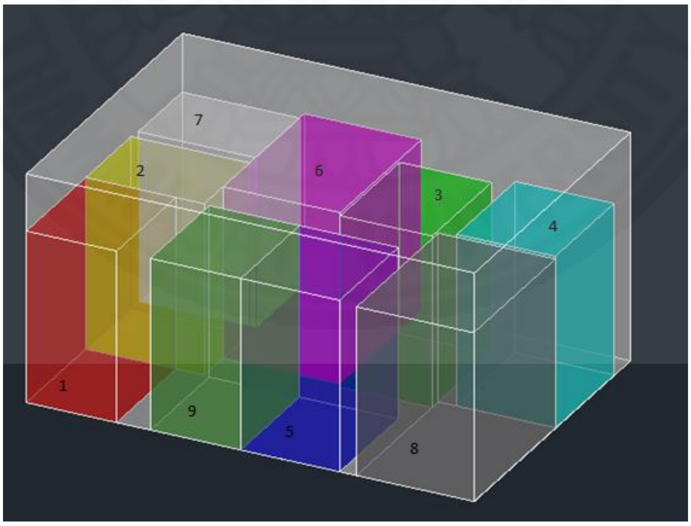


Figure 3.7. 9 cargos in ULD 1

Figure 3.7 was drawn with 3DAutoCad software 2015 from information in Table 3.9. Figure 3.7 gives image of positions of 9 cargos in ULD 1

Bellow tables are testing results for other three unsolved data sets with 20 cartons and 3 ULDs, and five data sets with 40 cartons and 5 ULDs. They were reported unsolved in Table 3.5 by original 3D packing model.

Table 3. 12 Bisection search results for data set 3 with 3 ULDs x 20 cartons

Iteration	Min Value %	Mid Point %	Max Value %	Selected Range	$\mid p_n - p_{n-1} \mid - \varepsilon$
1	0	50	100	Upper	
2	50	75	100	Lower	0.20196
3	50	62.5	75	Upper	0.07696
4	62.5	68.75	75	Lower	0.014462
5	62.5	65.625	68.75		-0.01679

Iteration	Objective Function		Time	Gap	Number of cartons packed
1	19620033	Feasible	00:00:08:06	0.00%	20/20
2		Infeasible	00:08:44:41	-	-
3	19620033	Feasible	00:00:09:10	0.00%	
4		Out of time	01:00:00:00	-	-
5	19620033	Feasible	00:00:08:80	0.00%	20/20

Table 3.12 provides testing results for data set 3 of 3ULDs and 20 cartons which was reported with gap 14.21% in Table 3.5 by original 3D packing model. As can be seen from the Table 3.12, acceptable optimal parameter can be found at iteration 5 with 65.625% to guarantee optimal solution without any constraint violated.

Table 3.13 provides testing results for data set 3 of 3ULDs and 20 cartons which was reported with gap 9.18 % in Table 3.5 by original 3D packing model. At iteration 5, acceptable optimal parameter is found at 78.125%.

Table 3.14 provides testing results for data set 3 of 3ULDs and 20 cartons which was reported “Out of memory” in Table 3.5 by original 3D packing model. As can be seen in Table 3.14, after 4 iterations, acceptable optimal parameter is found at 68.75 %.

Table 3. 13 Bisection search results for data set 4 with 3 ULDs x 20 cartons

Iteration	Min Value %	Mid Point %	Max Value %	Selected Range	$\| p_n - p_{n-1} \| - \varepsilon$
1	0	50	100	Upper	
2	50	75	100	Upper	0.18764
3	75	87.5	100	Lower	0.06264
4	75	81.25	87.5	Lower	0.00014
5	75	78.125	81.25		-0.03111

Iteration	Objective Function		Time	Gap	Number of cartons packed
1	17251913	Feasible	00:00:11:28	0.00%	18/20
2	15226630	Feasible	00:00:07:64	0.00%	20/20
3		Out of time	01:00:00:00	-	-
4		Infeasible	00:59:50:25	-	-
5	15226630	Feasible	00:00:08:74	0.00%	20/20

Table 3. 14 Bisection search results for data set 5 with 3 ULDs x 20 cartons

Iteration	Min Value %	Mid Point %	Max Value %	Selected Range	$\| p_n - p_{n-1} \| - \varepsilon$
1	0	50	100	Upper	
2	50	75	100	Lower	0.185778
3	50	62.5	75	Upper	0.060778
4	62.5	68.75	75	Upper	-0.00172

Iteration	Objective Function		Time	Gap	Number of cartons packed
	17275040	Feasible	00:00:15:31	0.00%	17/20
		Infeasible	00:00:22:13	-	-
	15318954	Feasible	00:00:09:59	0.00%	19/20
	14655369	Feasible	00:00:11:09	0.00%	20/20

Table 3. 15 Bisection search results for data set 1 with 5 ULDs x 40 cartons

Iteration	Min Value %	Mid Point %	Max Value %	Selected Range	$\mid p_n - p_{n-1} \mid - \varepsilon$
1	0	50	100	Upper	
2	50	75	100	Lower	0.189524
3	50	62.5	75	Upper	0.064525
4	62.5	68.75	75	Lower	0.002025
5	62.5	65.625	68.75	Upper	-0.029225

Iteration	Objective Function		Time	Gap	Number of cartons packed
1	28629810	Out of time	01:00:00:00	0.20%	-
2		Infeasible	00:11:20:59	-	
3	21495933	Feasible	00:00:55:45	0.00%	39/40
4		Infeasible	00:06:59:57	-	-
5	20157600	Feasible	00:00:38:62	0.00%	40/40

As can be seen from Table 3.15, acceptable optimal parameter can be set at 65.625%, resulting from 5 iterations.

Table 3.16 provides testing results with bisection search for data set 2 of 40 cartons and 5 ULDs. As can be seen from the table, 8 iterations need to run until the acceptable optimal parameter is found at 69.14063 %.

Testing results for data set 3 with 40 cartons and 5 ULDs are given in Table 3.17. At iteration 5, the acceptable optimal parameter is found at 65.625 %. All cartons are packed into ULDs with the constraints satisfied.

From table 3.18, at iteration 5, the acceptable optimal parameter is found at 65.625 %. All cartons are packed under the set of constraints.

Table 3.19 provides the results for data set 5 with 40 cartons to be packed within 5 ULDs. After iteration 4, the acceptable optimal parameter is found at 68.75 %. All cartons are packed under the set of constraints.

Table 3. 16 Bisection search results for data set 2 with 5 ULDs x 40 cartons

Iteration	Min Value %	Mid Point %	Max Value %	Selected Range	$\lvert p_n - p_{n-1} \rvert - \varepsilon$
1	0	50	100	Upper	
2	50	75	100	Lower	0.196593
3	50	62.5	75	Upper	0.071593
4	62.5	68.75	75	Upper	0.009093
5	68.75	71.875	75	Lower	-0.02216
6	68.75	70.3125	71.875	Lower	-0.03778
7	68.75	69.53125	70.3125	Lower	-0.04559
8	68.75	69.14063	69.53125		-0.0495

Iteration	Objective Function		Time	Gap	Number of cartons packed
1	28629810	Out of time	01:00:00:00	0.33%	-
2		Infeasible	00:01:20:50	-	-
3	24493959		00:00:14:96	0.00%	40/40
4	24493959		00:00:50:93	0.00%	40/40
5		Infeasible	00:01:48:04	-	-
6		Infeasible	00:01:05:70	-	-
7		Infeasible	00:01:05:27	-	-
8	24493959	Feasible	00:00:50:20	0.00%	40/40

Table 3. 17 Bisection search results for data set 3 with 5 ULDs x 40 cartons

Iteration	Min Value %	Mid Point %	Max Value %	Selected Range	$\lvert p_n - p_{n-1} \rvert - \varepsilon$
1	0	50	100	Upper	
2	50	75	100	Lower	0.186557
3	50	62.5	75	Upper	0.061557
4	62.5	68.75	75	Lower	-0.000943
5	62.5	65.625	68.75	Upper	-0.032193

Iteration	Objective Function		Time	Gap	Number of cartons packed
1	28636510	Out of time	01:00:00:00	0.15%	-
2		Infeasible	00:05:05:20	-	-
3	21739833	Feasible	00:00:40:45	0.00%	39/40
4		Infeasible	00:09:20:62	-	-
5	19793495	Feasible	00:00:50:25	0.00%	40/40

Table 3. 18 Bisection search results for data set 4 with 5 ULDs x 40 cartons

Iteration	Min Value %	Mid Point %	Max Value %	Selected Range	$\lvert p_n - p_{n-1} \rvert - \varepsilon$
1	0	50	100	Upper	
2	50	75	100	Lower	0.188767
3	50	62.5	75	Upper	0.063767
4	62.5	68.75	75	Lower	0.001267
5	62.5	65.625	68.75	Upper	-0.029983

Iteration	Objective Function		Time	Gap	Number of cartons packed
1	28629810	Out of time	01:00:00:00	0.13%	-
2		Infeasible	00:08:01:45	-	-
3	21472410	Out of time	01:00:00:00	0.14%	-
4		Infeasible	00:00:13:21	-	-
5	19692467	Feasible	00:00:29:91	0.00%	40/40

Table 3. 19 Bisection search results for data set 5 with 5 ULDs x 40 cartons

Iteration	Min Value %	Mid Point %	Max Value %	Selected Range	$\| p_n - p_{n-1} \| - \varepsilon$
1	0	50	100	Upper	
2	50	75	100	Lower	0.197825
3	50	62.5	75	Upper	0.072825
4	62.5	68.75	75	Upper	0.010325
5	68.75	71.875	75	Lower	-0.020925

Iteration	Objective Function		Time	Gap	Number of carton packed
1	28629810	Out of time	01:00:00:00	0.15%	-
2		Infeasible	1:00:00	-	-
3	25250165	Feasible	00:00:14:97	0.00%	40/40
4	25250165	Feasible	00:00:21:76	0.00%	40/40
5		Infeasible	00:50:00:42	-	-

3.4 Chapter Summary

This chapter deals with packing model for air cargo loading. Cartons for packing in the model have different sizes and released dates. The first model considers overlapping problem, practical position of cartons, cargo priority, weight limitation and time constraints. Testing results from Cplex shows the variation of runtime required for different configurations. For some configurations, no optimal solution is obtained within the CPU memory of the computer, in reasonable time. An idea to relax some constraints was proposed and tested. The relaxed model provides solution with the same data set which was reported unsolved within reasonable time from the original 3D packing model. The relaxed model also takes into account of priority of cargos when not all cargos are necessary to be packed, but chosen based on their priorities.

Chapter 4
Air Shipment Assignment Model

4.1 Mathematical model

In this section, an air shipment assignment model is presented. In general, the capacity is fixed by contracts for a given time period; however, to support the forwarder's handling cargo, airlines allow ad-hoc booking which is beyond allotment contract and charged with spot market prices. Ad-hoc bookings give opportunity to forwarder to access available extra capacity to load their cargos. The prices are subject to market demand at the time of booking, therefore the prices are probably higher or lower than contracted prices. Therefore, the decisions are subject to numbers of ULDs offered by airlines. Although the forwarder can cancel their allotment with an airline in written notification 24 hours prior to departure time without penalty, the forwarder in this study is required to reserve minimum weight per week specified in the contracts. This is to help airlines reduce no-show rate. Besides, in order to guarantee enough time for handling cargo at airport, assigned ULDs should arrive at airport before departure time about 6 hours for freighter aircraft or 4 hours for passenger aircraft. Cargo must be air-freighted no more than two days of releasing, which means assignment decision, along with document, must be done within 2 days so that the shipment can depart on airplane in time. This research assumes that all decision must be made by 17:00 every day (closed official working hour). Airlines offer the forwarder several price policies. In this study, to simplify the model, all airlines offer quantity discount to the forwarder represented by a piecewise function. The policy can be explained that airlines divide weight of shipments into multiple segments, represented by breaking points. There are L breaking-points from a_1 to a_l. Consider an airline j, there is an amount of money p_{jl} per kg for each segment $(a_{l-1}, a_l]$, $l \in \{1..L\}$ according to allotment contracts and p'_{jl} which based on market demand. Figure.4.1 explains the policy graphically.

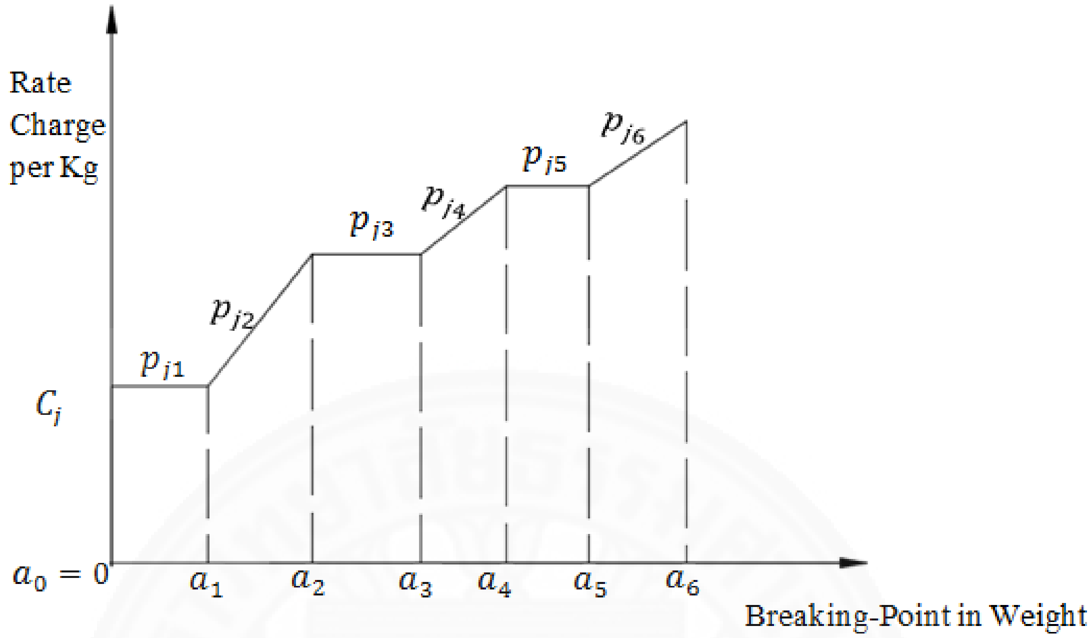


Figure 4. 1 An example of piecewise function for air cargo charges

For example, an airline could have a policy as shown in Table 4.1

Table 4.1 An example of price a policy from an airline

From a_{l-1} to a_l (Kg)	0-100	100-300	300-500	500-1000	1000-3000
p_{j1} (\$)	0	3	0	1.5	0

From Table 4.1, with the fixed cost equal to 500 US Dollars/shipment, a shipment with 900 Kg costs 500 + 0. 100+ 3.(300-100) + 0. (500 - 300) + 1.5. (900 – 100 – 200 - 200) = 1700 US Dollars.

Parameters:

N: a set of contracted airlines

U: a set of ULDs used to load cargo

T: A set of working day per week

L: A set of breaking-point in weight stated by airlines

C_j : Fixed cost to rent ULDs from airline j

q_{jt}: Maximum number of ULDs to be booked from airlines j on day t

p_{jl} : Contracted rate of segment l by airlines j

p'_{jl}: Rate on spot market of segment l by airlines j

s_j: Co-efficient representing the type of aircraft used

w_i : Weight of ULD i

m_j: Minimum weight per week required by airline i

$\mathrm{d_{jt}}$: Departure time of airline i on day t

$\mathrm{r_i}$: Ready day of ULD i

o_{i}: Earliest day of carton in ULD i

a_l: Breaking-point weight in price policy

$\mathrm{g_{il}}$: Weight of ULD i distributed in the range ($\mathrm{a_{l-1,}}$ $\mathrm{a_l}$]

In the example of shipment 900 Kg, $\mathrm{g_{il}}$ values are shown as bellow:

Table 4. 2 Example of weight distribution for a shipment of 900 kg

From a_{l-1} to a_l (Kg)	0-100	100-300	300-500	500-1000	1000-3000
g_{il} (Kg)	100	200	200	400	0

M: Very large number

D: Maximum time from released date to departure date

Decision Variables:

Con_{ijt} : is equal to 1 if ULD i is assigned to airline j on day t, on contract basis, otherwise 0.

Ad_{ijt}: 1 if forwarder makes any ad-hoc booking for ULD i with airline j on day t, otherwise, 0.

Objective:

Minimize:

$$\sum_{i=1}^{U}\sum_{j=1}^{N}\sum_{t=1}^{T}(Cont_{ijt} + Ad_{ijt}).C_j +$$

$$\sum_{i=1}^{U}\sum_{j=1}^{N}\sum_{l=1}^{L} p_{il}.g_{il}.Cont_{ijt} +$$

$$\sum_{i=1}^{U}\sum_{j=1}^{N}\sum_{l=1}^{L}\sum_{t=1}^{T} p'_{jlt}.g_{il}.Ad_{\ ijt}$$

Objective is to minimize total costs of renting ULDs, and variable costs, which depend on actual weights of ULDs and piece-wise prices charged by airlines. This function considers ad-hoc booking made by the forwarder in case the forwarder's demand exceeds contracted capacity or when the forwarder would like to gain benefit from difference in contracted price and spot market price.

<u>Constraints:</u>

a. General constraints:

$$\sum_{j=1}^{N}\sum_{t=1}^{T} Cont_{ijt} + \sum_{j=1}^{N}\sum_{t=1}^{T} Ad_{ijt} = 1 \quad \forall\ i\ \in U \tag{1}$$

Constraint 1 ensures that every ULD must be assigned only once in the planning horizon.

b. Specific constraints

Those are constraints relating to commitments between the forwarder, airlines, and their customer.

$$\sum_{i=1}^{U} Ad_{ijt} \le q_{jt} \qquad \forall\ j \in N, t \in T \tag{2}$$

$$\sum_{i=1}^{U}\sum_{t=1}^{T} (Cont_{ijt} + Ad_{ijt}).w_i \quad \ge m_j \quad \forall j \in N \tag{3}$$

$$d_{jt} - r_i \ge s_j - M.(1 - Cont_{ijt} - Ad_{ijt}) \ \ \forall\ i \in U,\ j \in N,\ t \in T \tag{4}$$

$$d_{jt} - o_i \ge D + M.(1 - Cont_{ijt} - Ad_{ijt}) \ \ \forall\ i \in U,\ j \in N,\ t \in T \tag{5}$$

Constraint (2) is the agreement between the forwarder and airlines about the maximum ad-hoc booking allowed for each working day. Constraint (3) guarantees minimum weight per week, which the forwarders must reserve with each contracted airline. Constraint (4) makes sure that cargos are ready early enough for handling at the airport. Constraint (5) ensures every cargo must be air freighted within D days after their released date.

4.2 Computational Results

This section present test results of shipment assignment model with several data sets. The data sets are of airlines, serving route Hanoi- Charles de Gaulle, that have contract with the forwarder. The flight schedules are shown in Table 4.3.

The first column indicates Charles De Gaulle, Paris, France (CDG) as the destination of shipments. Airline column shows two-character designation codes assigned by the International Air Transport Association (IATA) to all airlines. In the column, respective airlines are Vietnam Airlines, Korean Air Lines, Qatar Airways, Singapore Airlines, Cathay Pacific Airways, Emirates, and Hong Kong Airlines (HX). There are two types of aircrafts, which require different delivery time of shipment at the airport. The first represents passenger aircraft (P) which is used mainly to serve passengers; however, the available belly space of aircraft could be used to serve cargos. The second represents freighter (F), which is used for transporting cargos only. Shipment to be air freighted must be available at the airport before departing time 4 hours for passenger aircraft and 6 hours for freighter. The next seven columns present the flight schedule with aircraft codes respectively.

The forwarder signs contracts with airlines for specified route for each day; however, airlines still allows limited ad-hoc booking to support the forwarders in case the number of shipments is beyond allotment capacities or the spot market price is remarkable lower than the contracted prices. Table 4.4 shows the number of ad-hoc booking allowed.

Table 4. 3 Flight schedule of contracted airline for route HAN-CDG

Destination	Airline	Type	Mon	Tue	Wed	Thu	Fri	Sat	Sun
CDG	VN	P		VN019	VN019	VN019	VN019	VN019	VN019
				23:35	23:35	23:55	23:35	23:35	23:35
	KE	F	KE8362	KE8362		KE388	KE362	KE8362	
			0:55	4:15		6:15	3:15	23:00	
						KE0380		KE0372	
						5:45		2:00	
		P	KE480	KE480	KE480	KE480	KE480	KE480	KE0480
			12:25	12:25	12:25	12:25	12:25	12:25	12:20
	QR	P	QR835	QR835	QR835	QR835	QR835	QR835	QR835
			16:45	16:45	16:45	16:45	16:45	16:45	16:45
		F			QR8951	QR8951	QR8951	QR8951	QR8951
					8:55	20:00	21:50	20:40	21:00
	SQ	P	SQ175	SQ175	SQ175	SQ175	SQ175	SQ175	SQ175
			12:40	12:40	12:40	12:40	12:40	12:40	12:40
	CX	F		CX048	CX2048	CX048	CX2048	CX048	CX3248
				21:45	20:35	21:45	20:05	21:45	22:50
	EK	F				EK9897		EK9897	
						2:15		2:15	
	EK (via HX)	P		HX9018	HX9018		HX9018	HX9018	HX9018
				22:25	21:20		21:30	22:25	23:50

Table 4. 4 The number of ad-hoc booking allowed

	Working day						
	Mon	Tue	Web	Thu	Fri	Sat	Sun
VN019	0	2	2	2	2	2	2
KE8362	0	0	2	2	2	2	3
KE480	0	0	1	1	1	1	2
QR835	0	0	1	1	1	1	1
QR8951	0	0	2	0	2	0	0
SQ175	0	0	1	1	1	1	1
CX048	0	0	2	2	0	2	2
EK9897	0	0	0	2	0	2	0
HX9018	1	1	1	1	1	1	1

As can be seen from the Table 4.4, the number of ad-hoc booking can be different among airlines, and among operation days.

Table 4. 5 An instance of rule to generate random data for prices

	Breaking Points (Kg)					
	0-100	100 - 300	300 - 500	500 - 1000	1000 - 3000	3000 - 5000
Random Between (USD)	0	3.5- 5	0	2.5 - 3.4	0	1 - 2.3

Parameter D in the air shipment assignment model, which states possible maximum time from released date of shipment until department, is equal to 48 hours (2 days) plus 7 hours from 17:00 to midnight of the day. Since the information regarding to price policies is treated confidentially, the data for testing problem instances is generated randomly with excel, which is described as follow.

- Fixed costs for renting ULDs for each airline are random between 500 to 700 US dollars.
- There are six breaking points in chargeable weight stated by airlines. The contracted and ad-hoc prices for each airline are subject to piecewise function, and random value, but adhere to IATA's list of airfreight rates where the greater

the chargeable weight of the shipment, the cheaper the unit price. An instance of rule to generate the random data for prices is presented in Table 4.5

- The minimum weights required by airlines are random between 2000- 3500 (Kgs)

Table 4. 6 Contracted Rate and required minimum weight

	Breaking Points						Min Weight
	0-100	100 - 300	300 - 500	500 - 1000	1000 - 3000	3000 - 5000	
VN019	0	3.78	0	2.94	0	2.18	2457
KE8362	0	3.57	0	3.17	0	1.15	2652
KE480	0	4.05	0	3.20	0	1.92	2548
QR835	0	4.01	0	3.37	0	1.42	3237
QR8951	0	4.61	0	3.33	0	2.08	2646
SQ175	0	4.71	0	2.87	0	1.89	2018
CX048	0	4.74	0	3.00	0	1.49	2931
EK9897	0	3.88	0	2.97	0	1.22	2711
HX9018	0	4.90	0	2.84	0	1.54	2455

Table 4.6 provides information on contracted freight rate (transportation cost), and required minimum weight per week, which are fixed along contract period but is various among airlines.

Table 4. 7 Spot Market rate of Vietnam airlines for VN019

VN019	Mon	Tue	Web	Thu	Fri	Sat	Sun
0-100	0	0	0	0	0	0	0
100-300	4.52	3.58	4.05	3.98	3.92	4.92	4.13
300-500	0	0	0	0	0	0	0
500-1000	2.99	2.55	2.91	3.37	3.18	2.75	2.64
1000-3000	0	0	0	0	0	0	0
3000-5000	2.28	2.08	1.38	1.26	1.65	1.87	2.29

Table 4. 8 Spot Market rate of Korean Air Lines for KE8362

KE8362	Mon	Tue	Web	Thu	Fri	Sat	Sun
0-100	0	0	0	0	0	0	0
100-300	3.56	4.64	4.90	4.28	4.82	4.75	4.30
300-500	0	0	0	0	0	0	0
500-1000	2.87	2.88	3.31	2.65	2.93	3.17	3.25
1000-3000	0	0	0	0	0	0	0
3000-5000	1.51	1.14	1.99	1.43	2.18	1.50	1.02

Table 4.9 Spot Market rate of Korean Air Lines for KE480

KE480	Mon	Tue	Web	Thu	Fri	Sat	Sun
0-100	0	0	0	0	0	0	0
100-300	4.16	4.30	4.26	3.71	4.95	3.74	4.45
300-500	0	0	0	0	0	0	0
500-1000	2.58	3.16	3.17	2.99	3.20	2.66	3.39
1000-3000	0	0	0	0	0	0	0
3000-5000	1.28	1.14	1.63	2.20	1.94	1.75	1.88

Table 4.10 Spot Market rate of Qatar Airways for QR835

QR835	Mon	Tue	Web	Thu	Fri	Sat	Sun
0-100	0	0	0	0	0	0	0
100-300	4.52	3.58	4.05	3.98	3.92	4.92	4.13
300-500	0	0	0	0	0	0	0
500-1000	2.99	2.55	2.91	3.37	3.18	2.75	2.64
1000-3000	0	0	0	0	0	0	0
3000-5000	2.28	2.08	1.38	1.26	1.65	1.87	2.29

Table 4.11 Spot Market rate of Qatar Airways for QR8951

QR8951	Mon	Tue	Web	Thu	Fri	Sat	Sun
0-100	0	0	0	0	0	0	0
100-300	3.80	4.00	4.39	4.47	3.62	3.69	4.74
300-500	0	0	0	0	0	0	0
500-1000	3.18	2.77	2.90	2.71	3.19	2.91	2.73
1000-3000	0	0	0	0	0	0	0
3000-5000	2.17	1.10	1.52	2.06	1.47	1.58	2.17

Table 4.12 Spot Market rate of Singapore Airlines for SQ175

SQ175	Mon	Tue	Web	Thu	Fri	Sat	Sun
0-100	0	0	0	0	0	0	0
100-300	3.67	4.35	4.47	4.54	3.74	4.50	4.43
300-500	0	0	0	0	0	0	0
500-1000	2.67	2.68	2.83	2.93	2.81	3.38	3.06
1000-3000	0	0	0	0	0	0	0
3000-5000	1.37	1.69	1.71	2.16	1.28	1.41	1.78

Table 4. 13 Spot Market rate of Cathay Pacific Airways for CX048

CX048	Mon	Tue	Web	Thu	Fri	Sat	Sun
0-100	0	0	0	0	0	0	0
100-300	4.52	3.58	4.05	3.98	3.92	4.92	4.13
300-500	0	0	0	0	0	0	0
500-1000	2.99	2.55	2.91	3.37	3.18	2.75	2.64
1000-3000	0	0	0	0	0	0	0
3000-5000	2.28	2.08	1.38	1.26	1.65	1.87	2.29

Table 4. 14 Spot Market rate of Emirates for EK9897

EK9897	Mon	Tue	Web	Thu	Fri	Sat	Sun
0-100	0	0	0	0	0	0	0
100-300	3.84	3.80	3.72	4.30	3.53	3.55	4.55
300-500	0	0	0	0	0	0	0
500-1000	2.64	3.20	2.87	2.90	2.87	2.54	3.27
1000-3000	0	0	0	0	0	0	0
3000-5000	1.78	2.10	1.44	2.01	2.23	1.04	1.07

Table 4. 15 Spot Market rate of Hong Kong Airlines for HX9018

HX9018	Mon	Tue	Web	Thu	Fri	Sat	Sun
0-100	0	0	0	0	0	0	0
100-300	4.79	4.68	4.48	3.72	4.96	4.88	3.73
300-500	0	0	0	0	0	0	0
500-1000	3.21	2.55	3.25	3.06	3.40	3.06	3.16
1000-3000	0	0	0	0	0	0	0
3000-5000	2.25	1.26	2.08	1.76	1.14	2.21	1.74

Tables 4.7- 4.15 give information about spot market rate of 9 airlines which have flight schedule shown in Table 4.3. As can be seen from the tables, the rates change from time to time and are subject to market demand.

For this illustrated instance, ULDs to be assigned are a set of results from packing model with an assumption that all ULDs are supposed to be available at airport at 17:00 of the planned date.

Table 4.16 An example of ULDs to be assigned

	Weight	Earliest Release of cargos in ULD	Ready day of ULD
ULD1	4001	20- Jan	21- Jan
ULD2	3744	21- Jan	21- Jan
ULD3	2972	21- Jan	23- Jan
ULD4	3977	22- Jan	22- Jan
ULD5	3395	22- Jan	24- Jan
ULD6	3384	22- Jan	23- Jan
ULD7	3498	23- Jan	23- Jan
ULD8	3430	23- Jan	23- Jan
ULD9	2841	24- Jan	26- Jan
ULD10	2829	24- Jan	25- Jan
ULD11	2879	25- Jan	25- Jan
ULD12	2515	25- Jan	26- Jan
ULD13	2820	25- Jan	26- Jan
ULD14	2815	26- Jan	26- Jan
ULD15	2700	26- Jan	26- Jan

Information in Table 4.16 contains 15 ULDs, which are air shipments to be assigned. This test took around thirteen seconds (00:00:13:71) with an optimal solution (48836.95 US dollar). This model has 1977 constraints and 1891 variables. The assignment results are shown in Table 4.17.

Table 4. 17 Assignment results from Cplex

	Airlines	Departure day	Contract	Non-Contract
ULD1	QR835	22-Jan	1	0
ULD2	EK9897	23-Jan	1	0
ULD3	CX048	23-Jan	1	0
ULD4	KE8362	24-Jan	1	0
ULD5	HX9018	24-Jan	1	0
ULD6	SQ175	24- Jan	1	0
ULD7	EK9897	25- Jan	0	1
ULD8	EK9897	25- Jan	0	1
ULD9	VN019	26- Jan	0	1
ULD10	QR8951	26- Jan	1	0
ULD11	KE480	26- Jan	1	0
ULD12	VN019	26- Jan	0	1
ULD13	VN019	26- Jan	1	0
ULD14	VN019	26- Jan	1	0
ULD15	VN019	26- Jan	1	0

Table 4.17 shows how to assign 15 air shipments among airlines at either contracted price or non-contracted price (ad-hoc price) so that the total transportation cost is minimized and no commitment between airlines, the forwarder, and customers is violated.

Different sets of data are also selected randomly to test the model. Table 4.18 presents the testing results of other instances.

Table 4. 18 Testing results of other instances with Cplex

Data Set	Constraint	Variables	Optimal Results	Time	Gaps
3 Airlines x 10 ULDs					
1	454	421	29585.51883	00:00:01:40	0%
2			31495.33536	00:00:01:34	0%
3			30776.6599	00:00:01:32	0%
3 Airlines x 15 ULDs					
1	669	631	43893.17016	00:00:02:61	0%
2			44877.06542	00:00:03:22	0%
3			43404.97355	00:00:02:64	0%
3 Airlines x20 ULDs					
1	884	841	43284.47355	00:00:02:54	0%
2			58687.0097	00:00:02:43	0%
3			58443.2463	00:00:02:63	0%
3 Airlines x 30 ULDs					
1	1314	1261	87501.79168	00:00:02:86	0%
2			89362.25776	00:00:03:59	0%
3			88161.13535	00:00:03:08	0%
6 Airlines x 10 ULDs					
1	898	841	32506.16668	00:00:02:08	0%
2			32372.65477	00:00:02:83	0%
3			33750.02961	00:00:02:54	0%
6 Airlines x 15 ULDs					
1	1323	1261	48842.6827	00:00:01:77	0%
2			44454.83562	00:00:02:87	0%
3			47414.26906	00:00:03:60	0%
6 Airlines x 20 ULDs					
1	1748	1681	60998.05368	00:00:03:84	0%
2			61106.40468	00:00:04:11	0%
3			62200.78826	00:00:03:02	0%
6 Airlines x 30 ULDs					
1	2598	2521	88830.17729	00:00:03:08	0%
2			89629.58033	00:00:04:72	0%
3			88307.96833	00:00:04:73	0%
9 Airlines x 10 ULDs					

1	1342	1261	32175.87339	00:00:01:95	0%
2			32433.58018	00:00:02:49	0%
3			34006.60289	00:00:01:85	0%
9 Airlines x 15 ULDs					
1	1977	1891	32321.56428	00:00:02:92	0%
2			31652.57572	00:00:02:91	0%
3			32595.48666	00:00:02:38	0%
9 Airlines x 20 ULDs					
1	2612	2521	60624.69113	00:00:03:18	0%
2			59014.41919	00:00:04:58	0%
3			56973.85021	00:00:02:87	0%
9 Airlines x 30 ULDs					
1	3882	3781	87818.15554	00:00:03:44	0%
2			89600.69875	00:00:03:47	0%
3			89665.37627	00:00:03:97	0%

Table 4.18 shows that the model took a short time to run. During generating random data relating to airlines and ULDs, there are some guidelines to the forwarder in choosing carrier for allotment contracts. The first important factor is the departure time. Since urgent shipments, whose due dates to be air freighted are close, are possibly available at any time, the forwarder must reserve capacity from airlines with flight time between 17:00 to 24:00 every day. Another notice is about minimum weight. In cases, the number of airlines is relatively large compared with the number of ULDs. Minimum weight per week should not be high; otherwise, the model would encounter conflict between constraints about departure time and minimum weight requirement.

4.3 Chapter 4 summary

This chapter presents air shipment assignment model mathematically and testing results of model with different data sets. The model aims to assign shipments among contracted airlines with (or without) contracted rate. The objective of the model is to minimize the total transportation cost, which is formulated as piecewise function. Practical constraints related to handling cargos at airport and commitments between the forwarder and their customers are taken into account in the model. Air shipment assignment model is the second and the final part of loading planning. Since other constraints, related to limited volume, weight are considered in air cargo 3D packing model previously, this model gets simple and takes short time to run. However, the model is flexible to consider other practical constraints. Also, during the building and testing model, some guidelines for the forwarders in choosing carrier for allotment contracts are discussed in this chapter. The combination of two models fulfills the decision process of shipping cargo by the forwarder in reasonable time.

Chapter 5
Conclusion

This research presented two models to deal with air cargo loading management at a typical logistics forwarder that has allotment contracts with several airlines for their single-leg air shipment. The first model is air cargo 3D packing model, the second is air shipment assignment model.

Air cargo 3D packing model addresses loading shipments of multiple-sizes and released dates into ULDs in order to minimize unused space of ULDs. The model considers overlapping problem, practical position of cartons, weight limitation and other commitments stated by the forwarder. Several data sets with different configurations were generated to test the air cargo 3D packing model. For some configurations, the model generated solution in very short time; however, there were some cases where no solution was obtained within reasonable time and computer memory. A relaxed model was introduced to deal with unsolved cases. The relaxed model was also extended to consider the case where not all cargos are necessary to be packed into ULDs but chosen based on their priorities.

Air shipment assignment model supports the forwarder to assign their air shipment among contracted airlines either at contracted or non-contracted prices. The objective of the model is to minimize the total transportation cost, which is represented as piecewise function. Practical constraints related to handling cargos at airport and commitments between the forwarder and their customer are taken into account in the air shipment assignment model. Since other constraints, related to limited volume, weight are considered in air cargo 3D packing model previously, this model gets simple and takes short time to run. The research suggested some guidelines for the forwarders in choosing carrier for allotment contracts.

- The forwards must reserve capacity from airlines with flight time between 17:00 to 24:00 every day to handle urgent shipments whose dues to be shipped are close.

- Minimum weight required should be strongly considered when the number of contracted airlines is relatively large compared to the number of forecasted shipment.

Finally, the research opens for future studies to consider other practical constraints which can be considered in both models. Also, the research can be extended to solve the problems of forwarders who have to manage allotment contracts for multiple-leg air shipments.

References

Articles

A. Bortfeldt, & G. Waescher. (2013). Constraints in container loading-a state-of-the-art review. *European Journal of Operational Research, 229*, 1-20.

A. Lodi, S. Martello, & Vigo, D. (2002). Recent advances on two-dimensional bin packing. *Discrete Applied Mathematics, 123*, 379 - 396.

B. Slager, & Kapteijns, L. (2003). Implementation of cargo revenue management at KLM. *J. Revenue Pricing Manage, 3*, 80-90.

Baltacioglu, E. (2001). *The Distributor's Three Dimentional Pallet Packing Problem: A Human Intelligence- Based HeuristicApproach.* (Master of Science), Air Force Institute of Technology, Ohio.

C. Paquay, Schyns, M., & Limbourg, S. (2014). A mixed integer programming formulation for the three-dimensional bin packing problem deriving from an air cargo application. *International Transactions in Operational Research, 00*, 1-27.

C.S. Chen, S.M. Lee, & Q.S. Shen. (1995). An analytical model for the container loading problem. *European Journal of Operational Research, 80*, 68-76.

D.I. Patsiatzis, & L.G. Papageorgiou. (2002). Optimal multi-floor process plant layout. *Computers and Chemical Engineering, 26*, 575-583.

E. Hadjiconstantinou, & Christofides, N. (1995). An exact algorithm for general, orthogonal, two-dimensional knapsack problems. *European Journal of Operational Research, 83*(1), 39-56.

Huang, K., & Chi, W. (2007). A Lagrangian relaxation based heuristic for the consolidation problem of airfreight forwarders. *Transportation Research Part C: Emerging Technologies, 15*(4), 235-245.

I. Castillo, & Westerlund, T. (2005). An ε-accurate model for optimal unequal-area block layout design. *Computers & Operations Research, 32*, 429-447.

J. Beasley. (1985). An exact two-dimensional non-guillotine cutting tree search procedure. *Operations Research, 33*(1), 49-64.

J. Herz. (1972). Recursive computational procedure for two-dimensional stock cutting. *IBM Journal of Research and Development, 16*, 462-469.

J. Westerlund, L. Papageorgiou, & Westerlund, T. (2005). A Problem Formulation for Optimal Mixed-Sized Box Packing. *European Symposium on Computer Aided Process Engineering – 15*.

J.S. Billings, A.G. Diener, & B.B. Yuen. (2003). Cargo revenue optimization. *J. Revenue Pricing Manage., 2,*, 69-79.

K. Amaruchkul, W. L. Cooper, & Gupta, D. (2011). A note on air-cargo capacity contracts. *Production and Operations Management, 20*(1), 152-162.

L. Junqueira, R. Morabito, & Yamashita, D. S. (2012). MIP-based approaches for the container loading problem with multi-drop constraints. *Annals of Operations Research, 199*, 51-75.

L.G. Papageorgiou, & G.E. Rotstein. (1998). Continuous-domain mathematical models for optimal process layout. *Industrial and Engineering Chemistry Research,, 35*, 1354-1361. .

Leung, L. C., Van Hui, Y., Wang, Y., & Chen, G. (2009). A 0-1 LP Model for the Integration and Consolidation of Air Cargo Shipments. *Operations Research, 57*(2), 402-412.

Li, Y., Tao, Y., & Wang, F. (2009). A compromised large-scale neighborhood search heuristic for capacitated air cargo loading planning. *European Journal of Operational Research*, *199*(2), 553-560.

Li, Z., Bookbinder, J. H., & Elhedhli, S. (2012). Optimal shipment decisions for an airfreight forwarder: Formulation and solution methods. *Transportation Research Part C: Emerging Technologies*, *21*(1), 17-30.

M. Padberg. (2000). Packing small boxes into a big box. Mathematical Methods of Operations Research. *52*, 1-21.

N. Christofides, & C. Whitlock. (1977). An algorithm for two-dimensional cutting problems. *Operations Research*, *25*(1), 30-44.

P.C. Gilmore, & Gomory, R. E. (1965). Multistage cutting problems of two and more dimensions. *Oper. Res.*, *13*, 94-119.

R. Hellermann. (2006). Capacity Options for Revenue Management: Theory and Applications in Air Cargo Industry. *Springer*.

R. Tsai, E. Malstrom, & Kuo, W. (1993). Three dimensional palletization of mixed box sizes. *IIE Transactions 14*, *25*(4), 64-75.

S. Limbourg, M. Schyns, & Laporte, G. (2012). Automatic aircraft cargo load planning. *Journal of the Operations Research Society*, *63*(9), 1271-1283.

Wong, W. H., Leung, L. C., & Hui, Y. V. (2009). Airfreight forwarder shipment planning: A mixed 0-1 model and managerial issues in the integration and consolidation of shipments. *European Journal of Operational Research, 193*(1), 86-97.

Xue, J., & Lai, K. K. (1997). A study on cargo forwarding decisions. *Computers and Industrial Engineering*, *33*, 63-66.

Online Document

Boeing. (2014). *World Air Cargo Forecast 2014-2015*. Retrived July 15, 2015, from http://www.boeing.com/resources/boeingdotcom/commercial/about-our-market/cargo-market-detail-wacf/download-report/assets/pdfs/wacf.pdf

Appendices

Appendix A

AIR CARGO 3D PACKING MODEL

```
/* Parameters */


int NumCartons =... ;
int NumULDs = ...;
float M = 100;

range N = 1..NumCartons;
range O = 1..NumULDs;



float p[N] = ... ; /*length of carton*/
float q[N] = ... ; /*width of carton*/
float r[N] = ... ;  /*height of carton*/
float we[N] = ... ; /*weight of carton*/
float v[N] = ... ;            /*release date of carton*/
float pr[N] = ... ;     /*priority level of carton*/

float L = 317 ;  /*Length of ULd*/
float W = 223 ;  /*width of ULD*/
float H = 162 ;  /*height of ULD*/


/* Decision variables */
dvar boolean S[N][O];    /* if carton i is assigned to ULDj*/
dvar boolean n[O];        /* if ULD j is used*/


dvar float+ x[N] ;
dvar float+ y[N] ;
dvar float+ z[N]  ;

dvar boolean  lx[N]   ;     /* length of carton i is parallel to x-
axis*/
dvar boolean  ly[N];
dvar boolean  lz[N]   ;

dvar boolean wx[N];       /* width of carton i is parallel to x-axis*/
dvar boolean wy[N]   ;
dvar boolean wz[N] ;

dvar boolean hx[N]     ; /*height of carton i is parallel to x-axis*/
dvar boolean hy[N];
dvar boolean hz[N] ;

dvar boolean a[N][N]  ;      /* box i is on the left of box j */
dvar boolean b[N][N]  ;    /*right*/
dvar boolean c[N][N]  ;      /*behind*/
dvar boolean d[N][N]  ;      /*in front*/
dvar boolean e[N][N]  ;      /*below*/
dvar boolean f[N][N]  ;  /*above*/

dvar float  TotalVolume[O] ;
dvar float UsedVolume[O];
dvar float m[O];

dexpr float Space [j in O] = sum ( j in O) L*W*H*n[j] - sum (i in N,
j in O) p[i]*q[i]*r[i]*pr[i];

/*********************************************** model
************************************************/

minimize sum (j in O) Space[j];

subject to
{

  forall( i in N)
 Ct1:
sum(j in O) S[i][j] == 1;
/* (2) */

  forall (j in O)
 Ct2:
 sum (i in N) we[i]*S[i][j] <=4500;
                                          /*(3)*/

  forall (j in O, i in N, k in N: i <k)
 Ct3:
    abs (v[k]-v[i]) <= 2+ M*(1-S[i][j])+M*(1-S[k][j]);
      /*4*/

  forall (j in O)
   m[j]== sum (i in N)we[i]*S[i][j]; /*5*/

  forall( j in O)
   Ct5:
     sum(i in N) S[i][j] <= M*n[j];      /*6*/

  forall (j in O)
 { TotalVolume[j] ==  L*W*H*n[j];                           /*7*/
 UsedVolume[j] == sum(i in N)(p[i]*q[i]*r[i]*S[i][j]) ;}    /*8*/


  forall( i in N, k in N: i <k)
  {

  Ct9:

    x[i] +p[i]*lx[i] + q[i]*wx[i]+ r[i]*hx[i] <= x[k] + (1 -
a[i][k])*L ;       /* (9) */
  Ct10:
    x[k] +p[k]*lx[k] + q[k]*wx[k] + r[k]*hx[k] <= x[i] + (1 -
b[i][k])*L ;       /* (10) */
  Ct11:
    y[i] +q[i]*wy[i]+ p[i]*ly[i] + r[i]*hy[i] <= y[k] + (1-c[i][k])*W
;       /* (11) */
  Ct12:
   y[k]+q[k]*wy[k] + p[k]*ly[k] + r[k]*hy[k] <= y[i] + (1-d[i][k])*W
;        /* (12) */
  Ct13:
    z[i]+r[i]*hz[i] + q[i]*wz[i] + p[i]*lz[i] <= z[k] + (1-e[i][k])*H
;        /* (13) */
  Ct14:
    z[k] +r[k]*hz[k] + q[k]*wz[k] + p[k]*lz[k] <= z[i] + (1-
f[i][k])*H ;        /* (14) */

  }
  forall( i in N, k in N)
   {z[i]<= f[i][k]*H ;         /*15*/
   z[k] <= e[i][k]*H;}               /*16*/
  forall( i in N, j in O, k in N: i <k)
   Ct17:
   a[i][k] +b[i][k] + c[i][k]+ d[i][k] +e[i][k] + f[i][k] >= S[i][j]
+ S[k][j] -1;        /* (17) */


  forall( i in N, j in O)
  {

  Ct18:
    x[i] + p[i]*lx[i] + q[i]* wx[i] + r[i]* hx[i] <= L + (1-
S[i][j])*M ;                  /* (18) */
  Ct19:
    y[i] + q[i]*wy[i] + p[i]* ly[i] + r[i]*hy[i] <= W + (1-S[i][j])*M
;                   /* (19) */
  Ct20:
    z[i] + r[i]*hz[i] + q[i]*wz[i] + p[i]*lz[i] <= H + (1-S[i][j])*M
;                   /* (20) */
  }

forall (i in N)
  {lx[i]+ly[i]+lz[i]==1;
  wx[i]+wy[i]+wz[i]==1;
  hx[i]+hy[i]+hz[i]==1;
  lx[i]+wx[i]+hx[i]==1;
  ly[i]+wy[i]+hy[i]==1;
  lz[i]+wz[i]+hz[i]==1;}
};
```

Appendix B

RELAXED MODEL

```
/* Parameters */


int NumCartons =... ;
int NumULDs = ...;
float M = 1000;

range N = 1..NumCartons;
range O = 1..NumULDs;

float vol[N]=...;  /*volume of carton*/
float p[N] = ... ; /*length of carton*/
float q[N] = ... ; /*width of carton*/
float r[N] = ... ; /*height of carton*/
float we[N] = ... ;/*weight of carton*/
float v[N] = ... ; /*release date of carton*/
float pr[N]= ... ; /*priority level of carton*/


float L = 317 ;  /*Length of ULd*/
float W = 223 ;  /*width of ULD*/
float H = 162 ;  /*height of ULD*/
float Fd =...; /* maximum usage percentage of ULD*/


/* Decision variables */


dvar boolean S[N][O];    /* if carton i is assigned to ULDj*/
dvar boolean n[O];        /* if ULD j is used*/
dvar float  TotalVolume[O] ;
dvar float UsedVolume[O];
dvar float m[O];   /*weight of ULd after packing*/

/***************************model***************************/

minimize sum ( j in O) L*W*H*n[j] - sum (i in N, j in O)
vol[i]*S[i][j];

subject to
{
  forall( i in N)
 Ct1.1:
 sum(j in O) S[i][j] <= 1; /* All cartons must be assigned to ULDs*/

forall( i, k in N: i<k)
Ct1.2:
     maxl ((pr[k]-pr[i]), 0) <= maxl (( sum (j in O) S[k][j]*pr[k]-
sum (j in O) S[i][j]*pr[i]), 0);
```

```
 forall( j in O)
 Ct2:
 sum(i in N) vol[i]*S[i][j] <= L*W*H ;   /*Total volume of cartons
inside container cant exceed ULD volume*/

 forall (j in O)
 Ct3:
 sum (i in N) we[i]*S[i][j] <=4500;  /*weight limitation*/

  forall( j in O)
   Ct4:
 sum(i in N) S[i][j] <= M*n[j];     /* which ULDs are current in
usage*/

  forall (j in O, i in N, k in N: i <k)
  Ct5:
   abs (v[k]-v[i]) <= 2+ M*(1-S[i][j])+M*(1-S[k][j]);



 forall (j in O)
{ TotalVolume[j] ==  L*W*H*n[j];
UsedVolume[j] == sum(i in N)vol[i]*S[i][j];
 UsedVolume[j]<= Fd*TotalVolume[j];}

 forall (j in O)
 m[j]== sum (i in N)we[i]*S[i][j];
};
```

Appendix C

SCRIPT FOR MAIN FLOW CONTROL

```
main
 {
    thisOplModel.settings.mainEndEnabled = true;


    // new code
    var config_relax = new IloOplRunConfiguration("CargoRelax1.mod",
"CargoRelax1.dat");
      var model_relax = config_relax.oplModel;
    var data_relax = model_relax.dataElements;
    model_relax.generate();
    if ( config_relax.cplex.solve() ) {
      writeln("OBJ Relax = " + config_relax.cplex.getObjValue());
      ///writeln (Opl.use.solutionValue);
         writeln("Cargo Assignment Results" +
model_relax.S.solutionValue);

    }


    // Modify data items from relaxed model


    // Set packing model
    var src_packing = new IloOplModelSource("Packing.mod");
      var def_packing = new IloOplModelDefinition(src_packing);
      var opl_packing = new IloOplModel(def_packing,cplex);
    var data_packing = new IloOplDataSource("Packing.dat");
    opl_packing.addDataSource(data_packing);
    opl_packing.generate();



   // Loop through solving packing models (number of
ulds(containers))
          var def_tmp1 = opl_packing.modelDefinition;
          var data_tmp1 = opl_packing.dataElements;
          data_tmp1.NumULDs = 1;
          var num_cartons = 0;

          for( var u=1; u <= model_relax.NumULDs; u++)
          {
                  num_cartons = 0;
              for( var n=1; n <= model_relax.NumCartons; n++)
            {
                  if(model_relax.S[n][u] == 1)
                  {
                     num_cartons++;  // increment number of carton by
1
                     data_tmp1.p[num_cartons] = model_relax.p[n];
                     data_tmp1.q[num_cartons] = model_relax.q[n];
                     data_tmp1.r[num_cartons] = model_relax.r[n];
                     data_tmp1.we[num_cartons] = model_relax.we[n];
                     data_tmp1.v[num_cartons] = model_relax.v[n];
                     //data_tmp1.pr[num_cartons] = model_relax.pr[n];
                        }
            }
            data_tmp1.NumCartons = num_cartons;

              opl_packing = new IloOplModel(def_tmp1, cplex);
            opl_packing.addDataSource(data_tmp1);
            opl_packing.generate();
            if ( cplex.solve() ) {
                        writeln("OBJ " + u +"="+
cplex.getObjValue());
                        writeln("Coordinate");
                        writeln("x" +  opl_packing.x.solutionValue);
                        writeln("y" + opl_packing.y.solutionValue);
                        writeln("z" + opl_packing.z.solutionValue);
                        for( var i=1; i <= opl_packing.NumCartons;
i++){
                              if (opl_packing.lx[i]== 1){
                                    writeln("Length Along x " + i );
                                    };
                              if (opl_packing.ly[i]== 1){
                                    writeln("Length Along y " + i  );
                                    };
                              if (opl_packing.lz[i]== 1){
                                    writeln(" Length Along z " + i );
                              };
                              if (opl_packing.hx[i]== 1){
                                    writeln("Height Along x " + i );
                                    };
                              if (opl_packing.hy[i]== 1){
                                    writeln("Height Along y " + i );
                                    };
                              if (opl_packing.hz[i]== 1){
                                    writeln(" Height Along z " + i );
                              };
                              if (opl_packing.wx[i]== 1){
                                    writeln("width Along x " + i  );
                                    };
                              if (opl_packing.wy[i]== 1){
                                    writeln("width Along y " + i  );
                                    };
                              if (opl_packing.wz[i]== 1){
                                    writeln(" Width Along z " + i );

                                    };

                                    };

                        feasible = 1;
            }
            else
            {
                  writeln("Infeasible at uld " + u );

                  feasible = 0;
                  break;
            }

        }
            // New code ------------------------- end ---------------
-------------------------------------------------------

            if(feasible == 1) {
                  writeln("All Ulds feasible ");
                writeln("Script Ends");
  };


    // clear memory for packing model
      opl_packing.end();
      data_packing.end();
      def_packing.end();
      src_packing.end();


    // clear memory for relaxed model
    config_relax.end();


 }

/* Parameters */


int NumCartons =... ;
int NumULDs = ...;
float M = 1000;

range N = 1..NumCartons;
range O = 1..NumULDs;



float p[N] = ... ; /*length of carton*/
float q[N] = ... ; /*width of carton*/
float r[N] = ... ;  /*height of carton*/
float we[N] = ... ; /*weight of carton*/
float v[N] = ... ;            /*release date of carton*/
//float pr[N] = ... ;   /*priority level of carton*/

float L = 317 ;  /*Length of ULd*/
float W = 223 ;  /*width of ULD*/
float H = 162 ;  /*height of ULD*/


/* Decision variables */
dvar boolean S[N][O];   /* if carton i is assigned to ULDj*/
dvar boolean n[O];       /* if ULD j is used*/
dvar float+ x[N] ;
dvar float+ y[N] ;
dvar float+ z[N]  ;

dvar boolean  lx[N]   ;     /* length of carton i is parallel to x-
axis*/
dvar boolean  ly[N];
dvar boolean  lz[N]   ;

dvar boolean wx[N];      /* width of carton i is parallel to x-axis*/
dvar boolean wy[N]   ;
dvar boolean wz[N] ;

dvar boolean hx[N]     ; /*height of carton i is parallel to x-axis*/
dvar boolean hy[N];
dvar boolean hz[N] ;

dvar boolean a[N][N]  ;      /* box i is on the left of box j */
dvar boolean b[N][N]  ;    /*right*/
dvar boolean c[N][N]  ;      /*behind*/
dvar boolean d[N][N]  ;      /*in front*/
dvar boolean e[N][N]  ;      /*below*/
dvar boolean f[N][N]  ;  /*above*/

dvar float  TotalVolume[O] ;
dvar float UsedVolume[O];

dvar float m[O];

/********************************************* model
**********************************************/

minimize sum ( j in O) L*W*H*n[j] - sum (i in N, j in O)
p[i]*q[i]*r[i]*S[i][j];

subject to
{


  forall( i in N, k in N: i <k)
  {

  Ct1:
    x[i] +p[i]*lx[i] + q[i]*wx[i]+ r[i]*hx[i] <= x[k] + (1 -
a[i][k])*L ;       /* (1) */
  Ct2:
    x[k] +p[k]*lx[k] + q[k]*wx[k] + r[k]*hx[k] <= x[i] + (1 -
b[i][k])*L ;       /* (2) */
  Ct3:
    y[i] +q[i]*wy[i]+ p[i]*ly[i] + r[i]*hy[i] <= y[k] + (1-c[i][k])*W
;       /* (3) */
  Ct4:
   y[k]+q[k]*wy[k] + p[k]*ly[k] + r[k]*hy[k] <= y[i] + (1-d[i][k])*W
;       /* (4) */
  Ct5:
    z[i]+r[i]*hz[i] + q[i]*wz[i] + p[i]*lz[i] <= z[k] + (1-e[i][k])*H
;       /* (5) */
  Ct6:
    z[k] +r[k]*hz[k] + q[k]*wz[k] + p[k]*lz[k] <= z[i] + (1-
f[i][k])*H ;       /* (6) */

   }
   forall( i in N, k in N) // awkward position
    {z[i]<= f[i][k]*H ;
    z[k] <= e[i][k]*H;}
   forall( i in N, j in O, k in N: i <k)
    Ct7:
    a[i][k] +b[i][k] + c[i][k]+ d[i][k] +e[i][k] + f[i][k] >= S[i][j]
+ S[k][j] -1;       /* (7) */

forall( i in N)
 Ct8:
sum(j in O) S[i][j] == 1;
/* (8) */

   forall( j in O)
    Ct9:
      sum(i in N) S[i][j] <= M*n[j];
/* (9) */

   forall( i in N, j in O)
   {

   Ct10:

     x[i] + p[i]*lx[i] + q[i]* wx[i] + r[i]* hx[i] <= L + (1-
S[i][j])*M ;                  /* (10) */
   Ct11:
     y[i] + q[i]*wy[i] + p[i]* ly[i] + r[i]*hy[i] <= W + (1-S[i][j])*M
;                  /* (11) */
   Ct12:
     z[i] + r[i]*hz[i] + q[i]*wz[i] + p[i]*lz[i] <= H + (1-S[i][j])*M
;                  /* (12) */
   }

forall (j in O)
 { TotalVolume[j] ==  L*W*H*n[j];
 UsedVolume[j] == sum(i in N)(p[i]*q[i]*r[i]*S[i][j]) ;}


forall (i in N)
   {lx[i]+ly[i]+lz[i]==1;
   wx[i]+wy[i]+wz[i]==1;
   hx[i]+hy[i]+hz[i]==1;
   lx[i]+wx[i]+hx[i]==1;
   ly[i]+wy[i]+hy[i]==1;
   lz[i]+wz[i]+hz[i]==1;}

forall (j in O)
    m[j]== sum (i in N)we[i]*S[i][j];

};
```

Appendix D

AIR SHIPMENT ASSIGNMENT MODEL

```
//Index
int N=...;
range ULDs= 1..N;
{string} Airlines=...;
int t= ...;
range Workdays= 1..t;
int l=...;
range BreakingPoint= 1..l;

float MaxULD_Nu[Airlines][Workdays]=...;
float Fix_Cost[Airlines]=...;
float ContractedRate[Airlines][BreakingPoint]=...;
float SpotRate[Airlines][BreakingPoint][Workdays]=...;
float Aircraft_type[Airlines]=...;

float Dept[Airlines][Workdays]=...;

float w[ULDs]=...;
float r[ULDs]=...;
float o[ULDs]=...;
float a[BreakingPoint]=...;
float m[Airlines]=...;
float g[ULDs][BreakingPoint]=...;

//Decision variables

dvar boolean eContract[ULDs][Airlines][Workdays];
dvar boolean eNonContract[ULDs][Airlines][Workdays];


//Expression
minimize sum (i in ULDs, j in Airlines, t in Workdays)
(eContract[i][j][t]+ eNonContract[i][j][t]) * Fix_Cost[j] +
           sum (i in ULDs, j in Airlines, l in BreakingPoint, t in
Workdays)ContractedRate[j][l]*g[i][l]* eContract[i][j][t] +
             sum (i in ULDs, j in Airlines, l in BreakingPoint, t in
Workdays) SpotRate[j][l][t]*g[i][l]*eNonContract[i][j][t];

subject to {
 Ct1:
   forall (i in ULDs)
  sum (j in Airlines, t in Workdays)eContract[i][j][t]+ sum (j in
Airlines, t in Workdays) eNonContract[i][j][t]==1;  /*every ULD must
be assigned into only time*/

  Ct2:
    forall (j in Airlines, t in Workdays)
      sum (i in ULDs)eNonContract[i][j][t]<= MaxULD_Nu[j][t]; /* max
number of ULDs on spot market to be booked*/

  Ct3:
    forall (j in Airlines)
      sum (i in ULDs,  t in Workdays) eContract[i][j][t]*w[i]>= m[j];
/*min weight per week by airline*/
```

```
  Ct4:
    forall (i in ULDs, j in Airlines, t in Workdays)
      Dept[j][t]-r[i]>= Aircraft_type[j]- 1000* (1-
eContract[i][j][t]- eNonContract[i][j][t]);   /*show up time at
airport*/

  Ct5:
    forall (i in ULDs, j in Airlines, t in Workdays)
      Dept[j][t]- o[i] <= 48 + 7 + 1000*(1 - eContract[i][j][t]-
eNonContract[i][j][t]); /*must airfrieghted no more than 2 days after
released*/

} ;

execute DISPLAY {
   for(var i in ULDs)
      for(var j in Airlines)
      for(var t in Workdays)
      {
         if(eContract[i][j][t] == 1)
            write("eContract[",i,"][",j,"][",t,"] =
",eContract[i][j][t]);
         if(eNonContract[i][j][t] == 1)
            write("eNonContract[",i,"][",j,"][",t,"] =
",eNonContract[i][j][t]);}
          };
```