AN INTEGRATION OF CONTAINER LOADING AND VEHICLE ROUTING PROBLEMS

ECEM BARIŞ

AUGUST 2016

AN INTEGRATION OF CONTAINER LOADING AND VEHICLE ROUTING PROBLEMS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
IZMIR UNIVERSITY OF ECONOMICS

BY

ECEM BARIŞ

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE
OF
MASTER OF SCIENCE
IN
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

AUGUST 2016

Approval of the Graduate School of Natural and Applied Sciences

(Prof. Dr. İsmihan BAYRAMOĞLU)
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

(Prof. Dr. Mustafa Arslan ÖRNEK)
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Asst. Prof. Dr. Zeynep SARGUT
Supervisor

Examining Committee Members Date: 26.8.2016

Prof. Dr. Mustafa Arslan ÖRNEK

Asst. Prof. Dr. Zeynep SARGUT

Asst. Prof. Dr. Ceyhun ARAZ

ABSTRACT

AN INTEGRATION OF CONTAINER LOADING AND VEHICLE ROUTING PROBLEMS

BARIŞ, Ecem

M.Sc. in Industrial Engineering

Graduate School of Natural and Applied Sciences

Supervisor: Asst. Prof. Dr. Zeynep SARGUT

August 2016, 70 pages

Vehicle routing and container loading are the core functions of transportation. The integration of vehicle routing and container loading is becoming a research area for the logiscits companies. To get competitive advantage over other firms, firms should satisfy customer demand on time, with the right amount and high-quality service. By integrating the vehicle routing and container loading, firms can carry their products by less number of vehicles, which means reduced fuel cost but also time-reduction on deliveries. We focused on white-good industry in this thesis. Vehicle routing and container loading problems are NP-hard and very difficult to solve. In this thesis, a mathematical model was developed for the integrated capacitated vehicle routing problem with time windows and three-dimensional loading problem. Combination of these models are examined and then a decompostion method is proposed.

Keywords: time window, vehicle routing, loading, three-dimensional, mathematical modeling, decomposition, heuristic

ÖZ

KONTEYNER YÜKLEME VE ARAÇ ROTALAMA PROBLEMLERİNİN BİR ENTEGRASYONU

BARIŞ, Ecem

Endüstri Mühendisliği Yüksek Lisans Programı

Fen Bilimleri Enstitüsü

Tez Danışmanı: Yrd. Doç. Dr. Zeynep SARGUT

Ağustos 2016, 70 sayfa

Araç rotalama ve konteyner yükleme taşımacılığın çekirdek işlevlerindendir. Bununla birlikte, bu iki problemin entegrasyonu, şirketlerin sıklıkla araştırma alanı haline gelmiştir. Rakip firmalar üzerinde rekabet avantajı ve müşteri memnuniyeti sağlamak için firmalar, müşteri taleplerini zamanında, doğru miktarda ve kaliteli bir şekilde karşılamak zorundadırlar. Araç rotalama ve yükleme entegrasyonu sayesinde, firmalar ürünlerini daha az araçla taşır. Böylece daha düşük yakıt maliyeti ve daha kısa teslimat süresi elde ederler. Taşımacılıkta kullanılan ürünlerin özellikleri değişkenlik gösterdiğinden, bu tez konusu için beyaz eşya sektörü seçilmiştir. Araç rotalama ve yükleme sorunları NP zor olmakla birlikte, verimli bir şekilde çözülememektedir. Gerçek hayattaki kullanımlara uygunluğu test etmek amacıyla, belirlenen araç rotalama, zaman aralığı ve üç boyutlu yükleme problemi için bir matematiksel model geliştirilmiştir. Öncelikle iki ayrı problem birleşimi incelenip, sonrasında daha iyi bir çözüm sağlamak amacıyla ikiye bölünmüştür. Bu yaklaşımın avantajları tartışılmıştur.

Anahtar Kelimeler: zaman aralığı, araç rotalama, yükleme, 3 boyut, matematiksel modelleme, ayrıştırma, sezgisel

ACKNOWLEDGEMENTS

First of all, I would like to express my deepest gratitude to my supervisor Asst. Prof. Dr. Zeynep Sargut, who has supported me throughout my thesis and provided overall guidance of my work on this research. I wish to thank her for her guidance, contribution, useful comments, remarks and writing of this master thesis. Without her guidance and persistent help this master thesis would not have been possible.

I would like to express my gratitude to Dr. Cemalettin Öztürk for both his support and contribution in my BSc and MSc years. His guidance and experience always help my work in progress.

A special thanks to my friends İpek Süğüt and Burçin Ceren Olçum for their brilliant comments and suggestions on this research. I am very grateful for their support and encouragement.

Last but not the least important, I owe more than thanks to my family: my parents İrem and Mustafa Barış for giving birth to me, supporting me spiritually throughout my life and for their endless love. Without their support, it is impossible for me to finish my undergraduate and graduate programs seamlessly. I am very lucky to have such a great family.

Table of Contents

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1: INTRODUCTION

In today's business environment, Transportation Management (TM) is the core of the logistics processes. It includes scheduling, routing, loading and unloading operations. Most of companies are subcontracting transportation operations to logistic firms since the distribution of products is a vital operation for many companies.

The biggest advantage of this is; that they have some storage areas and the companies do not have to spend warehouse cost for it. Although, the logistic firms load the products to their vehicles and transport them to the customers, the production company is responsible for the due dates and customer satisfaction. Good transportation aims to deliver items on time, not to damage them and deliver the right amount. To satisfy the customer expectations, it is essential to integrate routing and vehicle packing. (Moura and Oliveira, 2009)

A 3-PL (3rd Party Logictics) company provides outsourced logistics services for the customers. Shipper can include warehousing, transportation, freight negotiation, reporting, forecasting in their contracts. This study is aimed to solve the problem of a 3-PL logistics company, which stores and distrubutes white goods directly to the retailers. This logistics company rents trucks which are used whole day to satisfy white good retailers' demands. Moreover, white goods have high volume and weight, it also effects the capacity utilized by the trucks. Only the routing is determined before the white goods are loaded. If the capacity of the truck is not enough to carry all goods of the retailer, then a new truck is arranged. They do not have an integrated system which combines both vehicle routing and truck loading problem. In this study, we aim to provide a combined solution which minimize the number of trucks and maximize the number of white goods loaded while all demand is satisfied.

This problem is related to two well-defined combinatorial optimization problems in the literature: vehicle routing problem and container loading problem. First, the number of vehicles and route of the each vehicle are determined; according to the solution, items of the each customer are combined into one group of items in the vehicle. If the vehicle capacity is not enough new vehicle is used and the routes are again created. In the first part of the problem VRP and CLP are examined separately, with their mathematical models; in the second stage, two problems considered as a one mathematical model which models the 3-PL company needs'. After that a heuristic will be developed for this combined model.

Now, two combinatorial optimazation problem are introduced separetly with their objectives, constraints, variations and importance.

1.1 Vehicle Routing Problem

Vehicle Routing Problem (VRP) is a NP- hard combinatorial optimization problem. The roots of VRP come from Traveling Salesman Problem (TSP). TSP consists of one vehicle and a set of cities; which starts from depot and turns back to the depot. In VRP, there is more than vehicle; a number of vehicles located at a central depot have to serve a set of geographically dispersed customers. (Fig.1.)

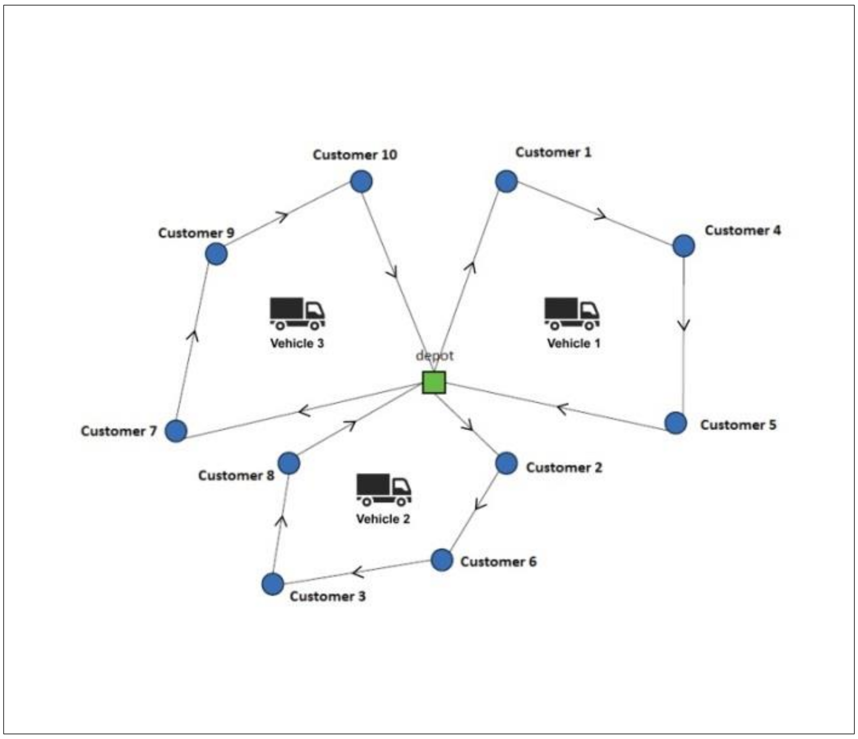


Figure 1 Vehicle Routing Problem

For example, vehicle 1 starts its route at the depot and then visits the forth customer, then fifth customer and again turns back to the depot, where it started to its tour. A number of vehicles satisfy all of the customer demands and sub-tour is not allowed. Sub-tour is a tour that one node visited more than one time in the route. The aim of the VRP is to serve a set of customers, with a known demand, minimizing the transportation cost, starting its route from the depot and each vehicle turns back to the depot. Each vehicle has a given capacity and each customer has a given demand.

VRP is a combinatorial optimization problem that means the number of feasible solutions increases exponentially when the number of customers increases (Bell & McMullen, 2004). There are many objectives of VRP such as minimization of global transportation cost (variable + fixed costs), minimization of the number of vehicles and balancing of the routes for travel time and vehicle load.

There are different variants of VRP according to their features. These are Capacitated VRP (CVRP), VRP with Time Windows (VRPTW), VRP with Backhauls (VRPB) and VRP with Pickup and Delivery (VRPDP). One of the most studied version is CVRP. The vehicles are identical has the same capacity and homogenous. Split deliveries are not allowed. However, CVRP is not enough for the loading part of the problem. It only takes in to account of the total weight of the load that does not exceed the given capacity of the vehicles.

VRPTW is another important problem for the logistics. Objective is as same as VRP, minimizing the number of vehicles and total distance traveled. The routes must be developed in such a way that each customer is visited only once by exactly one vehicle within a given time interval. Each route has to start and finish within the time window associated with the depot. Each vehicle has a capacity. Real life examples are bus routing, home delivery and technical services, food distributions.

Before starting CVRPTW, we given the notation below. The mathematical model is based on Desrochers et al. (1988)

Parameters:

$i, j, l \in N = \{0, \ldots, n+1\}$	N is set of nodes. Nodes "0" and "n + 1" represent depot, where every route must start and end, respectively. Nodes "1" to "n" represent n clients that must be visited. G is the graph G= (N,A).
$k \in K = \{1,2,\ldots v\}$	set of vehicles that can be used
$a \in A = \{\langle i, j\rangle \in N \times N \mid i \neq j \wedge i < n+1 \wedge j > 0\}$	set of arcs of G
c_{ijk}	traversing time of arc $\langle i, j\rangle \in A$ by vehicle k: $c_{ijk} > 0$
$[a_i, b_i]$	earliest and latest starting time of serving node i (i.e., time window)
C	capacity of a vehicle in terms of weight
t_i	service time client i
$M_{ijk} = \max\{0, b_i + c_{ijk} + t_i - a_j\}$	Big $\boldsymbol{M}$

Decision variables:

x_{ijk}	equals to 1 if vehicle k serves client j immediately after client i and 0 otherwise, $i = 0, \ldots, n$, $j = 1, \ldots, n + 1$, $i \neq j$, $k \in K$
s_{ik}	time at which vehicle k begins serving client i, $i = 0, \ldots, n + 1, k \in K$, $s_{ik} \in R^+$

$$\text{Min} \sum_{k=1}^{v} \sum_{i=0}^{n} \sum_{j=1}^{n+1} c_{ijk} x_{ijk} \quad (1)$$

Subject to

$$\sum_{k=1}^{v} \sum_{j=1}^{n+1} x_{ijk} = 1 , \quad i = 1,\ldots, n \quad (2)$$

$$\sum_{j=1}^{n} x_{0jk} = 1 , \quad k = 1,\ldots,v \quad (3)$$

$$\sum_{i=0}^{n} x_{ilk} - \sum_{j=1}^{n+1} x_{ljk} = 0 , \quad l=1,\ldots, n \;\; k=1,\ldots,v \quad (4)$$

$$\sum_{i=1}^{n} x_{i(n+1)k} = 1, \quad k = 1,\ldots,v \quad (5)$$

$$s_{ik} + t_i + c_{ijk} - s_{jk} \leqslant M_{ijk} (1-x_{ijk}) , \quad \langle i, j\rangle \in A \;\; k=1,\ldots v \quad (6)$$

$$a_i \le s_{ik} \le b_i \quad i = 0, \ldots , n + 1 \; k = 1, \ldots , v \quad (7)$$

$$s_{ik} \ge a_i + \sum_{j=1}^{n+1} \max\{0, a_j - a_i + t_j + c_{jik}\} x_{jik} ,$$

$$i = 0, \ldots , n + 1 \; k = 1, \ldots , v \quad (8)$$

$$s_{ik} \leqslant b_i - \sum_{j=1}^{n+1} \max\{0, b_i - b_j + t_i + c_{ijk}\} x_{ijk} ,$$

$$i = 0, \ldots , n + 1 \; k = 1, \ldots , v \quad (9)$$

$$\sum_{i=1}^{n} d_i \sum_{j=1}^{n+1} x_{ijk} \le C \quad k= 1,\ldots,v \quad (10)$$

$$x_{ijk} \in \{0, 1\} \quad (11)$$

$$s_{ik} \ge 0 \quad (12)$$

Expression (1) gives the objective of minimizing the total cost. Constraints (2) ensure that each customer is assigned to one route and served by only one vehicle. Constraints (3) provide that each vehicle must start its route from the depot. There should be flow between the customers in the same route. Constraints (4) and (5) state that flow starts with the depot and end with the depot. The inequalities (6) ensure that a vehicle k cannot arrive at j before $t_i + c_{ijk}$ if it is travelling from city i to city j. Constraints (7) state that time windows for each city. The consistency of the time variables s_{ik} ensured through linearized constraints (8) and (9). Constraints (10) provide the vehicle capacity restriction. Constraint (11) and (12) enforce that x_{ijk} is binary variable, and

s_{ik} is a non-zero variable. This model was coded in OPL. (Appendix-1)

Vehicle Routing Problem with Backhauls (VRPPB) is also known as the line-haul and back-haul problem, which is an extension of CVRP. The customer set is partitioned into two subsets as line-haul customers which require a given quantity of product to be delivered and back-haul customers, which give quantity of inbound products, must be picked up. (Toth & Vigo, 2002) As a real-life application; supermarkets are line-haul customers and grocery suppliers are the backhaul customers. Transportation cost is saved by visiting the backhaul customer in the distribution route. Most of the problems, line haul customers have higher priority than the backhaul customer, since the empty jars or boxes do not need to be collected immediately.

In the VRPDP, a number of vehicles have to serve a number of transportation requests. For each request, origin and destination locations are specified. Origin location where it is to be picked up plus a pickup time window, and the destination location where it is to be delivered plus a delivery time window. The difference between the VRBP is, all deliveries have to be served before pick-ups can begin. However, VRPPD provides that goods are transported between the pickup and delivery points. (Fig.2) For example, milk bottles are transported while empty ones must be returned to the origin depot.

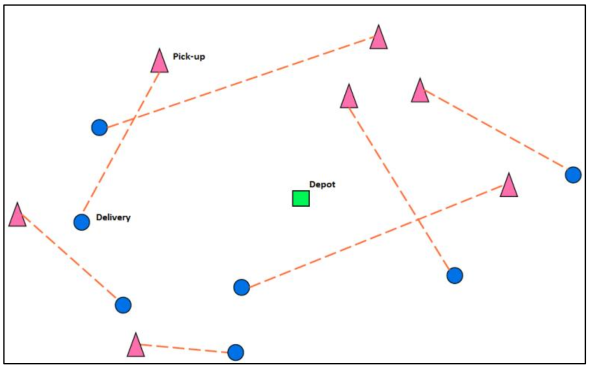


Figure 2 Vehicle Routing Problem with Pick-up and delivery

Since there are lots of different extensions of VRPs, there are also lots of solution procedures for VRP. These are linear programming, branch and bound, branch and cut, set covering based algorithms and finally heuristics such as Clarke and Wright, Cluster First Route Second and meta-heuristics. Clarke & Wright (1964) suggested two versions of the saving algorithm. First one is parallel savings algorithm and the other one is sequential savings algorithm. In the parallel version, more than one route may be built at a time; despite in the sequential version exactly one route is built at a time. A numerical example was used to explain differences between two heuristics clearly.

Let us consider 5 customers; with given transportation costs between all pairs, which are shown in the table, 0 refers the depot and the costs are symmetric.

Table 1 Transportation cost between customers, c_{ij}

From\To	0	1	2	3	4	5
0	-	28	31	20	25	34
1		-	21	29	26	20
2			-	38	20	32
3				-	30	27
4					-	25
5						-

Assume that the vehicle capacity is 100 units and the customer demands are shown in the below table.

Table 2 Customer demands

Customer	Quantity
1	37
2	35
3	30
4	25
5	32

By combining the two routes, we can obtain the savings as $S_{ij} = c_{i0}+c_{0j}-c_{ij}$.

For example, S_{12}=28+31-21=38.

The saving matrix is given in as Table 3.

Table 3 Saving (S_{ij}) matrix

i\j	1	2	3	4	5
1	-	38	19	27	42
2		-	13	36	33
3			-	15	27
4				-	34
5					-

The savings are sorted in descending order as following:
1-5, 1-2, 2-4, 4-5, 1-4, 3-5, 1-3, 3-4, 2-3.
Now, we can examine both savings algorithms after these calculations.

a) Sequential savings algorithm:

Since the highest saving is given by the 1-5, it is chosen firstly. Also capacity is checked, 69 units does not exceed the vehicle capacity, 100 units. Secondly, 1-2 is considered. This neighborhood can be provided by connecting 1-2-5 or 5-1-2. However, the total demand exceeds the vehicle capacity, and the 1-2 is not connected. 2-4 should be considered thirdly. 1-5 and 2-4 will be connected by more than one route but sequential savings only construct one route at a time. Next saving is the link 4-5, it can be connected 1-5-4 and total demand is not exceed vehicle capacity which means that it is feasible. According to the vehicle capacity no more nodes will be added to the route. 0-1-5-4-0 is the result for sequential saving. Total transportation cost is calculated as 98.

b) Parallel savings algorithm

Like in the sequential saving, it starts with 1-5. Since the parallel savings can build more than one route, 2-4 can be considered, and routes become 0-1-5-0 and 0-2-4-0. After that 3-5 is added to the first route; 0-1-5-3-0 and the total transportation cost is calculated as 171. For the parallel algorithm, after routes are built than they can be combined to reduce transportation cost if it is possible and it does not exceed.

In this thesis, the time-window has an important role since the products are delivered from warehouse to the retailers and each retailer has an agreement with the customer. Toth and Vigo (2002) formulated the mathematical model for CVRP with time-window constraints.

Heuristics are often greedy methods; they usually get trapped in a local optimum and thus fail. This heuristics are problem dependent. The other solutions method for VRP is metaheuristics which are problem-independent techniques, they are not greedy, and allow to explore more thoroughly the solution space and thus get a better solution with global optimum. Meta-heuristics are powerful techniques and applied generally for a large number of problems. A meta-heuristic is a set of algorithmic concepts that can be used to define heuristic methods applicable to a wide set of different problems.

Most meta-heuristics are inspired by a biological or physical phenomena such as simulated annealing (SA), tabu search (TS), genetic algorithm (GA), ant colony optimization algorithm (ACO) and particle swarm optimization (PSO). Most of them belong to the class of local search, which iteratively applying small local moves to a solution for finding a better one. Ant colony optimization (ACO) is a new technique among the other meta-heuristic methods. Firstly, Italian researchers (Dorigo, et al., 1996) developed the heuristic for simulating the food-seeking behaviors of ant colonies in the nature.

The vehicle routing problems are important for logistics firms; logistics companies usually neglect the dimensions of the loads; they only consider the capacity in weight and volume. Therefore; the vehicle routing problem could find the best route according the distance or cost but the loading could not be feasible at all times. For this reason, transportation companies should take loading constraints into account for realistic planning. The integration of routing and loading model has to be developed since many of the planning tools do not include these constraints.

1.2 Container Loading Problem

The shipping companies deal with the utilization of the container capacities. Maximizing the utilization of container space can affect the container usage and also it reduces the cost of shipment.

Container Loading Problem (CLP) is the problem, which loads a subset of rectangular boxes into a container with fixed dimension and maximizes the volume of the packed boxes. Furthermore, the container loading problem has different definitions on literature. Moreover, CLP problem with 3-dimensions are called 3-dimensional packing problem. It tries to fit different types of boxes in an optimal-level. (Gürbüz, et al. 2009) These types of problems aim to minimize the number of vehicles used.

Containers can be defined as vehicles, trucks and also items can be defined as boxes. It is a very complex combinatorial problem and it usually aims to maximize loading efficiency (Pisinger, 2002). The problem tries to hold two basic conditions except including the special requirements of the problem: all small items lie totally with in the container and the items should not be overlapped. (Borthfeldt & Wascher, 2012)

Dyckhoff, who was one of the earliest researchers of CLP, classified the problem into two types according to their container types; single container loading problem (SCLP) which is considered as one container and aim is to maximize the utilization of the container space and the second one is multiple container loading problem (MCLP) which considers more than one container (Dyckhoff, 1990). MCLP has two versions. The first one is bin packing problem (BPP) which tries to minimize the total number of containers used or minimize the total cost of vehicles used since the dimensions of the containers can be diffrent. A set of 3 dimensional items are packed into 3 dimensional rectangular boxes, which are refered as bins. The second one is Knapsack Loading Problem (KPP), when the available space in the container is not enough than the problem tries to maximize the total volume or the number of packed items. (Fig.3)

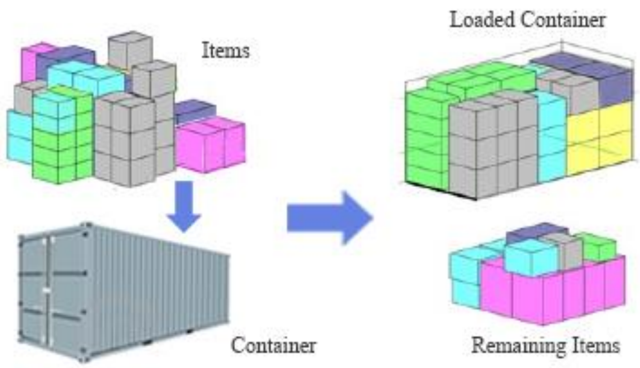


Figure 3 Knapsack Loading Problem (KPP)

Additionaly, Pisinger (2002) defines Strip Packing Problem (SPP) which tries to minimize the depth of the container while the width and the height of the container are fixed. Pisinger (2002) did not specify the BPP and KPP as a subcategory of MCLP, it can be also valid for SCLP. Also, if the containers are different in dimension and the items are strongly heterogeneous; it is called Multiple Heterogeneous Knapsack Problem (MHKP).

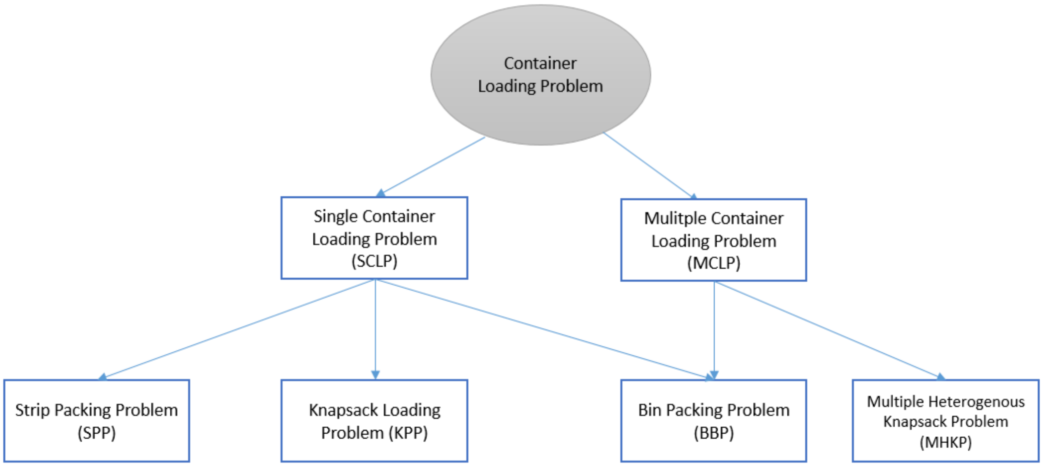


Figure 4 Classification of CLP

Wäscher et al. (2007) state that the container loading problems can be examined in terms of their input and output. Input problem types aim to minimize the number of used containers, which is in the category of BPP, and the objective of output problem types is maximizing number of loaded items. Moreover, SCLP and MCLP can be subcategorized by their item similarity. Container loading problems also classified based on the item type except the container amounts or dimensions. When the two items are identical then they have the same dimensions and if item set includes only one type, it means that items are homogenous. When the item type is not so large and the amount of item is high, it is called weakly heterogeneous. In contrast, if the item type is large and number of item is small then it is defined as strongly heterogeneous.

1.3 Integrated Vehicle Routing and Container Loading Problem

In transportation and distribution, vehicle routing problem (VRP) and container loading problem (CLP) are the two combinatorial problems which are related to each other.

In this study, we aim to solve VRPTW and CLP together. VRP decides which customers' package will be assigned to which vehicle so that the vehicle capacity is not exceeded. However, the arrangement of the boxes into the vehicle is not certain after solving VRP. Therefore, the problem became vehicle routing with time windows and loading problem. (VRTWLP)

In the literature, the four objectives are considered for VRTWLP: (1) minimization of the number of vehicles; (2) minimization of the total traveling distance; (3) minimization of the total time; and (4) the minimization of the total waiting time at clients.

There are some different cases, which play important role to define the problem. Borthfeldt and Wascher (2012) defined some of the requirements which can be considered as constraints for this real life problem for the CLP and VRP in the below.

1. **Orientation constraints:** White goods have rectangular shapes and according to their sensitivity, the orientation of loading is stable by the “This side up” sign, although some of the products can be flexible with more than one orientation. Selecting the one dimension as height, the vertical orientation of the box is defined. In two dimensional problems, mostly the longest measurement can be selected as length and the second longest measurement as a width. Vertical orientation constraints prevent the products and the packages being damaged. Moreover, Bischoff & Ratcliff (1995) discussed the horizontal orientation constraints, which is related with front and back side of the boxes. This constraint can either be defined in the mathematical model as a constraint or decided by the parameters of the model which is length, height or width.

2. **Load stability constraints:** During the transportation boxes can be damaged if the boxes cannot be supported by another item or side of vehicle. Unstable boxes may result in injuries while loading and unloading operations. These stability constraints can be vertical as well as horizontal. In vertical stability, items should be supported by another item or by the floor. It is related to the gravity. Junqueira et al. (2012) state that horizontal stability can be provided by lateral side of the boxes or with the container wall at least three of four sides. It is also known as dynamic stability. Liu et al. (2011) use a stability as a soft constraints; it only needs to contact with one side of the other box or the one side of the container.

3. **<u>Load bearing constraints</u>:** This is the maximum pressure which can be applied to the item since some of the items are placed above of it. (Junqueira et al., 2012) According to the type of items and their strength; some items can put on to another item. It is also related to its vertical orientation. Gendreau et al. (2006) mentioned the load bearing strength in a soft case. The boxes should be divided into two categories as fragile and non-fragile. Non fragile boxes can be put only over non-fragile boxes. However fragile boxes can be put only the non-fragile boxes. Furthermore, container loading problems should prohibit the large boxes being put on smaller ones. Since the items have different densities, some products are restricted to be the under of another product, or in contrast, some products should be at the bottom of the container. (Scheithauer and Terno, 1996)

4. **<u>Axle-weight constraints</u>:** These constraints should be used in large vehicles such as trucks and trailers. There are some axle-weight limits according to country. When an item placed in to a vehicle then the weight of item divided on to the axles of truck or axles of trailer. F_A^j shows the weight of items of customer j on axles of the trailer and the F_K^j is for the axles of the truck (Pollaris et al., 2014).

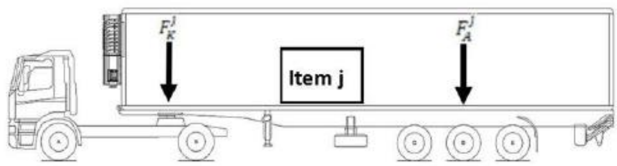


Figure 5 Axle weight truck and trailer

These constraints are also known as the weight distribution or load balance constrains. Weight of the items should be spread as possible as across the container floor (Gehring & Bortfeldt, 1997). If the container is balanced then it reduces the risk of shifting items while the truck moves.

5. **<u>Complete shipment of certain retailer order</u>:** Since the retailer orders many items, these products have to be shipped in the same vehicle. If the one items of the retailer are in the vehicle, then the other items have to be there. However, if one item of a customer cannot be loaded than the other items of the customer should not be loaded in the vehicle. This term is also defined as "*Mutual*

Positioning". These types of constraints are usually used in disassembled products like furniture. If the shipment is incomplete then there is no need to deliver the other parts of the complete item to the customer.

6. **<u>Multi-dimensional packing constraints</u>:** Each item in the vehicle cannot be overlapped and should be enclosed by the vehicle. This constraint is satisfied by checking each dimension separately.

7. **<u>Vehicle capacity</u>:** The capacity of the vehicles is not only considered in weight limits; there are also different vehicle types in terms of dimension and weight, which is called heterogeneous fleet. Moreover, when the number of city which will have visited per vehicle is low, and the container is weakly homogenous the final solution quality must be better than the other types of problems. If the dimensions of the products are with the less number of items, trucks are full and the number of city visited is also less.

8. **<u>Grouping same type of items</u>:** According to the space efficiency, it is better to group the item with the same type together, if they are also to be shipped to closer cities. Soft constraints can be used and some penalty costs can be given. Moreover, placing the items closely which belongs to the same customer, reduces the number of wrong unloading or loading operations of the items. (Haessler and Talbot, 1990).

9. **<u>Multi-drop items</u>:** Each retailer's products have to be closed to each other, unless there are lots of unloading and loading operations for the shipments. Moreover, each order of the client must be satisfied by a single vehicle. Each retailer's box positions in to the container must be packed together so that the unloading can be done easily. Furthermore, it is called unloading sequence constraints. It provides that when the deliveries have just arrived to a customer, no items belonging to that customer served later because of the blocking items of the other cities (Pollaris et al., 2014). Loading and unloading operations

must be done by LIFO (last-in-first-out) strategy. Moreover, in a one-dimensional packing problem it is called LIFO constraints.

10. **Shipment Priorities:** The boxes have high or low priorities according to the shelf-life of the products or deadlines and so on. Also Bortfeldt and Gehring (1999) divide priorities into two categories as soft and hard priorities. In hard loading priority, the boxes with high priorities exist in the container and the low ones may not exist if the high priorities are not almost inside of the container. In contrast, in soft loading priority, there are also low priority boxes in the container by using the coefficients in the objective function.

11. **Allocation Constraints:** When the items are shipped, the customer wants to receive all of his items in a single consignment. These allocation constraints are used in multiple container problems. On the other hand, some of the items like food or perfumery should not be transported in the same vehicle and these constraints are known as separation constraints. Liu et al. (2011) also mention allocation constraints as connectivity constraint, which tries to load the items in a particular subset in the same container.

CHAPTER 2: LITERATURE REVIEW

In the context of this thesis, we review the problems individually and then we review different integrations and applications of them. Literature will be analyzed in the following subcategories: VRP, CLP and the integration of CLP and VRP.

2.1 Vehicle Routing Problem

Toth and Vigo (2002) develop exact methods for VRP. Since the logistics problems include the reliability of the delivery; time-windows have to be considered. The starting time of the service and service time should be associated by the time-windows.

Moghaddam et al. (2006) aim to minimize the total fleet cost, routing cost and cost of violating the time-windows constraint. They consider a heterogeneous fleet. They restricted the customer satisfaction in the interval from a to b. (a and b is represented the bounds of soft time-window interval) If they do not deliver in this interval, they will be penalized with a penalty cost. Moreover; they define LB and UB for hard time windows. Service will not be allowed out of the hard time windows.

Since the Vehicle Routing problem is an NP-Hard combinatorial optimization problem and many heuristic methods are developed to solve it.

Ant Colony Optimization (ACO), introduced by Dorigo et al. (1996), has been applied to many optimization problems such as traveling salesman (Dorigo, et al., 1996), job-shop scheduling (Colorni et al., 1994) and quadratic assignment problems (Taillard, et al., 1997).

Bin et al. (2008) developed the Improved Ant Colony Optimization (IACO). This improved heuristic additionally updates the ant weight strategy, which means updating the increased pheromone. In another words, updating rule can integrate the global feature and the local feature, a mutation operation, and the 2-opt exchange for the VRP.

2.2 Container Loading Problem

The features of the containers and the items have a great impact on modeling since they are considered in real life. First of all, all related constraints are examined individually.

Hemminki et al. (1998) produced efficient and stable loads by an on-line packing algorithm. In this algorithm, only a single vertical orientation is allowed for the boxes which are marked as “this way up!” sign. Moreover, Doerner et al. (2010) mention this orientation constraint as the loading should be orthogonal, which means boxes must be loaded parallel to the side of vehicle and the box heights should be fixed. Since, this thesis considers white-good industry, the products have vertical orientation and the horizontal moves are allowed in the container.

Bischoff and Ratcliff (1995) apply many scenarios which can be used in container loading. Two different approaches are combined: load stability and multiple drop.

There are a lot of meta-heuristics used for CLP. In the literature, Bortfeldt et al. (2004) use local search algorithms and compare with other studies’ outcomes.

In the early 90’s, Haesseler & Talbot (1990) describe a heuristic for truck and rail loading problem and they also develop three-dimensional loading diagrams for the shipments of low-density products. Since there are lots of non-physical constraints, the development of an explicit mathematical model was impossible.

Osman et al. (2014) presented a loading model for a manufacturing company. Cylindrical parts have to put into the baskets. There are more than one layer and more than one basket. It aims to minimize the unutilized capacity of the baskets. They define the baskets in Cartesian coordinate system. The first model presented is nonlinear and aimed to reduce the complexity of three-dimensional loading problems. Moreover, two mixed integer models are developed: layer-loading problem and basket loading problem.

2.3 Integrated Vehicle Routing and Container Loading Problem

First of all, the extensions of VRP and CLP are widely explained in the Chapter 1. According to needs of white-good industry, this constraints are examined in more detail.

Allocation constraints are mostly used in the combination of vehicle routing and container loading problems by Moura & Oliveira (2009), Gendreau et al. (2006), Iori & Martello (2010). Moreover these constraints are used in a heuristic method for solving container loading problem in Eley (2003). The logistics companies do not want to travel the same retailer more than one time. Since there are multiple containers, loads may not fit into a single container. The objective of this kind of problems can be the minimization of the number of containers required.

Multi-drop condition is another important condition for our problem since each customer has subset of product groups; it may not affect the loading time of the item but it is important for the unloading conditions. Doerner et al. (2010) mention sequential loading in three-dimenisonal loading constraints. If the box b is visited after box a; then box a should not be at the top of item b.

Doerner et al. (2010) handled two well-known problems in combinatorial optimization CVRP and Bin packing problem (3BPP) by using ant colony optimization algorithm since it provides fast packing for the loading side of the problem. According to the problem, the related constraints are considered as below in 3L-CVRP (Three-Dimensional Loading Capacitated Vehicle Routing Problem).

In the literature, the combination of these two different problems are not studied much since each problem is hard by itself. Moreover, the real-world instances are also huge and the solutions are generally infeasible or do not converge to optimal solution in reasonable times. For this reason, some of the researchers examine the problems with 2-D loading constraints or 3-D loading constrains. VRP and two-dimensional loading problem combination are not studied in the literature too much. In VRP with 2-D Loading, the boxes and vehicle measurements are expressed in two dimensions, which is mostly width and length.

Pollaris et al. (2014) illustrate that the importance of the axle weight consraints in a VRP model. The model is used in a small network and for each route the weight on the axles are compared with the legal limits.

Leung et al. (2013) combined heterogeneous fleet vehicle routing problem with 2-dimensional loading constraint. In this article, they solve the vehichle routing problem with simulated annealing heuristic local search method and embeded packing heuristic for two dimensional loading part of the problem. The aim is to minimize the transportation cost and meet the customer demand. Transportation cost is calculated by the fixed and variable cost of the vehicle such as fuel consumption, also according to the vehicle dimensions, the consumption increases. Capacity of the vehicle is evaluated by the length and width dimensions; and split delivery is not allowed in this problem. For the loading side of the problem; heuristics of Zachariadis were used and all of them search for the most suitable position for each item.

In the 3L-CVRP, the dimensions of the boxes and vehicles are considered in three dimensions. Since the problem includes the height measurement then lots of feature can be added to the problem such as load bearing strength and stability of the boxes. The combination of VRP and CLP is another complex combinatorial problem since the objective of routing and loading are different. Although, the two problem types are widely studied in the literature, they are weakly combined in three-dimensional loading case.

Gendreau et al. (2006) introduce this integration with additional constraints and present a tabu-search heuristic. The model includes the weight capacity, load bearing, and orientation constraints.

Moura (2008) develop a multi-objective genetic algorithm to solve 3 dimensional VRP with time windows (3L-CVRP). The objective is to minimize the weighted sum of the number of vehicles and the total distance traveled.

Moura & Oliveira (2009) studied the model in Sixt (1996) and Martello et Al. (2000). They presented two different resolution methods: sequential and hierarchical approach. The sequential approach includes the simultaneous planning of routes and loading. In hierarchical approach, according to the feature of the loading: unloading sequence-based, orientation and stability constraints are taken into account. The objective is the sum of the number of vehicles and the total route time.

Some researchers assume 3D-packing problems with unlimited height for a container. Gürbüz et al. (2009) proposed a heuristic method to solve this kind of problems. The algorithm called Largest Area First-Fit (LAFF) and it aims to minimize the height of the container used. When the number of items increase, the complexity of the problem increases with non-deterministic polynomial time (Robinson, 1980). The algorithm works using two types of placement methods. First method tries to find the box with largest surface area by height minimization. After placing the boxes according to first method, the remaining boxes are allocated with the second method which fills the empty places.

All of the constraints are mentioned in Chapter 1 with their numbers. The combination of VRP and CLP are compared in the below table.

Table 4 Comparison table for constraints in the literature

	Constraints										
Literature	1	2	3	4	5	6	7	8	9	10	11
Doerner, Fuellerer, Gronalt, Hartl, & Iori (2010)	x	x	x		x		x		x		
Pisinger,(2002)						x	x				
Bischoff, & Ratcliff (1995)		x				x	x		x		
EB model (This study)	x				x	x	x				

CHAPTER 3: PROBLEM DEFINITION

Let us consider the problem where a set of customers demanding from a retailer, and the retailer should retain the products by the logistics firm. These logistic firms or the Third Party Logistics (3-PL) provider is the connection between buyer and seller (Fig.6).

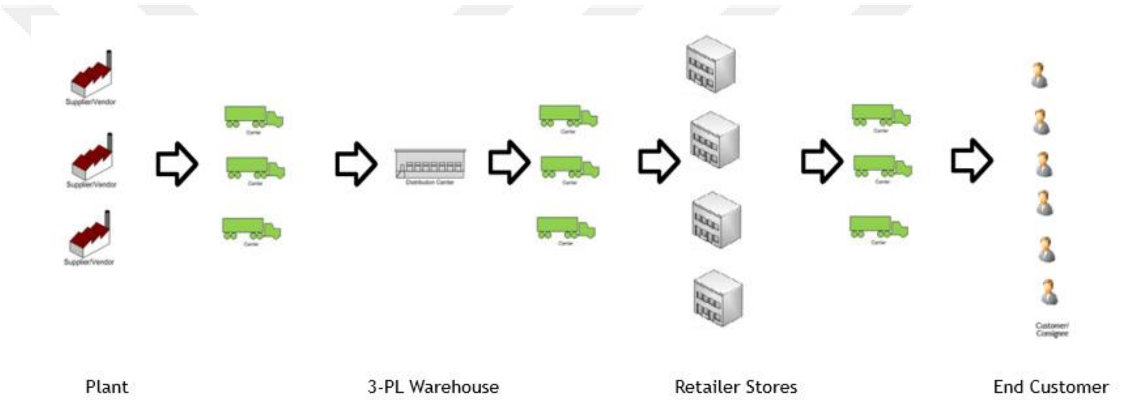


Figure 6 Flow for a 3-PL company

Normally, the loading operations done by the 3-PL companies are not managed professionally. It means that the employee which loads the container decided to put items according to their destinations (first city is loaded first, LIFO) or does not consider the distance only thinks about fitting as many as items as possible. In the problem, there is only one depot in Kemalpaşa and supplier produces goods, and the 3-PL is responsible for transporting them to the retailers. The problem starts from the loading operations at the warehouse and ends when all items are delivered to retailers.

Here are the flows of loading operations for one of the biggest 3-PL company in Izmir.

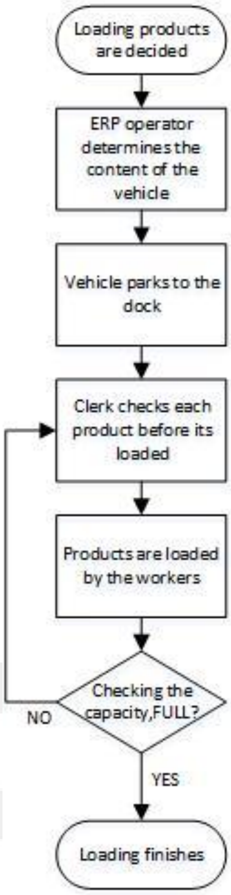


Figure 7 The flow chart of the loading operations

In the problem, goods are transported by the 3-PL firm and the vehicles are *heterogeneous*, it means they have different dimensions. There are lots of narrow streets in İzmir and not all of the types of the vehicle are used for the deliveries. Also, it is also efficient for the fuel consumption. Moreover, each cities' *demand* is *constant* and they are few different types, which are weakly heterogeneous.

Time Window: 3-PL companies are always charge of meet a deadline of the retailers' *time window*. This is the most common type of vehicle routing problem. Each retailer should be served between opening time to closing time and their time windows are different in terms of the social density.

Deci (cubic decimeter) Capacity: Except for the vehicle capacity, there is also capacity check for the white goods in terms of $dm^{3,}$ cubic decimeter. For the cargo and transportation companies, deci is used for volumetric, dimensional, weight calculation. In the following chapter, the calculation is shown.

Moreover, in Section 1.3, general attributes for the CLP were defined, which will be the main constraints of the problem. According to the needs of the 3-PL Company, the constraints of the problem are listed below.

Vehicle Capacity: Each vehicle has different *capacity* limit in terms of *weight*. According to Karayolları Trafik Yönetmeliği, weight capacity is limited by the government and it is calculated based on the number of tires and distance between the wheel bases.

Orientation constraint: As all know, the packages of the white goods include the “this side up” sign. It prevents the damage of the products for themselves. If the refrigerator is upside down then it can be harmed. Instead of using this orientation as a constraint, it can be written as parameter, if the dimensions are fixed.

Complete Shipment of certain retailer order: Each retailer demand is satisfied by a single vehicle, which means that split delivery is not allowed for the problem. All of the demands for one retailer must be *completed* in one *shipment*. Therefore, each retailer’s demand must be packed together to increase the efficiency of the unloading operations. (Moura & Oliveira, 2009) For example, if two vehicles satisfy the demand of one retailer, it should cause high fixed cost.

Multi-dimensional packaging constraint: Each dimension of the box should be checked to avoid overlapping in the vehicle.

Moreover, the problem has some assumptions;

- The number of vehicle and dimensions of each vehicle are known.
- Each vehicle’s weight capacity is known according to the rules and regulations of Turkey. (Karayolları Trafik Yönetmeliği)
- The dimensions and number of each box are known.
- Each dimension can be considered as width, height or depth according to how you view the box. The length of each box is placed at X-axis and the width of each box is placed at Y-axis and height of each box is placed at Z-axis.
- The back-right-bottom (BRB) corner of vehicle is fixed at origin.

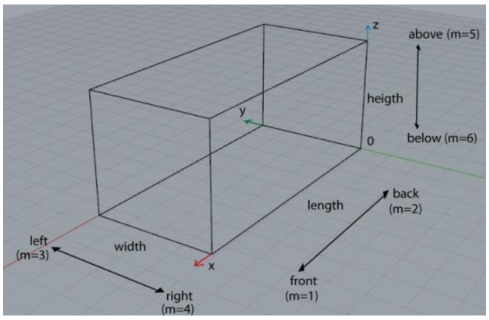


Figure 8 Directions of the boxes according to the dimensions

- Fuel cost is considered as variable cost.
- Average speed is defined in *Karayolları Genel Müdürlüğü Hız Sınırları* and fuel consumption (liters/100 km) data were provided from logistics company. Cost of diesel oil was calculated as 3,65 TRY/l iter.
- Vehicle rents are considered as fixed cost.
- The starting time is considered as 0 and the time unit is minutes.
- Service time of each retailer is directly proportional to the size of the demand. Each box is loaded in 3 minutes approximately.
- Covered vehicles are used, unloading operations can be done at the same time from the different side of the vehicle. Grouping in the truck of the items to each customer does not necessary.

Figure 9 Covered Vehicle

First of all, in order to develop a mathematical model, the input data sets are examined carefully. The inputs collected from the logistics company are the demand of each customer, number of vehicles, capacities of the vehicles and weight of the boxes, dimension of the vehicle and the boxes, time limitations of each customer (hard-time window and service time of each customer).

Since the time-window measurement is in minutes, the distance between cities (Republic of Turkey General Directorate of Highways, tarih yok) are converted from kilometers (km.) to minutes (min.). It should be calculated as below:

Table 5 Calculation of distance in minutes

$$Minutes\ between\ cities = \frac{\text{KM between cities}}{\text{Average KM per hour}}$$

Since the data belongs to the real-life, usually the rent of vehicles was calculated with the direct proportion of the type of the vehicles. There are two types of vehicle; light truck and medium truck.

Figure 10 Types of vehicles (Light Truck, Medium truck)

Fuel consumption is one of the important variables, which affects the objective function directly and related with the total distance for the vehicle.

Table 6 Calculation of fuel consumption in liter per minute

Fuel consumption (L/min) =Avg. speed (km/min)* fuel consumption (L/ km)

Time–window is determined by the opening and closing hours. The workers are available to load or unload the items during the working hours. These hours are based on the location of the retailer. Operating hours of a store is between 6 am and 10 pm.

CHAPTER 4: MATHEMATICAL MODEL

Since the real problem contains two different NP-Hard problems, we analyz them separately.

Moura and Oliveira (2009) presented a novel mixed integer programming model for simultaneous consideration of the VRPTW and CLP. While vehicle routing part of the model ensured that items are delivered to customers within the time windows, container loading aspect of the model formulation guaranteed a balanced load in which possible damages are prevented. Due to intractability of the developed model, authors did not attempt to solve the developed model.

An improved VRTW and CLP model can be formulated by introducing new variables that strengthen the model formulation.

G(N,A)	a directed graph, A is the set of arcs and N is the set of nodes *i, j, l* ϵ *N* ={*0,...,n, n+1*}:*G*'s set of nodes. Nodes "0" and "*n* + 1" represents the depot, where every route must start and end, respectively. Nodes "1" to "*n*" represent *n* clients that must be visited
k	vehicle index
α, α´, α´´	box indices
a	arc index, $a \in A = \left\{ \langle i, j \rangle \in N \times N \,\middle|\, i \neq j \wedge i < n+1 \wedge j > 0 \right\}$, be a set of arcs of *G*
f_i	weight of criterion *i* in the objective function based on their relative importance
c_{ijk}	traversing time of arc $\langle i, j \rangle \in A$ by vehicle *k*: $c_{ijk} > 0$

[$earliest_i$, $latest_i$]	earliest and latest starting time of serving node i (i.e., time window) [$earliest_i \leq latest_i$]
l_α, w_α, h_α	length, width and height (respectively) of box α
q_α	weight of box α
L_k, W_k, H_k	length, width and height (respectively) of vehicle k
Q_k	weight capacity of vehicle k
t_i	service time of vehicle k at client i
r_k	fixed cost associated with the use of vehicle k
m	$m \in Side = \{1,2,\ldots,\|6\|\}$ relative sides of boxes, where $m=1$ refers to being in the front, $m=2$ in the back, $m=3$ on the left, $m=4$ on the right, $m=5$ above or $m=6$ below of a box respectively.
$e(i)$	set of boxes belonging to node i
M_1, M_2	big numbers used in linearization of the logical constraints.

Decision variables:

$$x_{ijk} = \begin{cases} 1, & \text{if vehicle } k \text{ serves client } j \text{ immediately after client } i \\ 0, & \text{otherwise} \end{cases}$$

$$y_k = \begin{cases} 1, & \text{if vehicle } k \text{ is used} \\ 0, & \text{otherwise} \end{cases}$$

$$\gamma_{\alpha k} = \begin{cases} 1, & \text{if box } \alpha \text{ is transported by vehicle } k \\ 0, & \text{otherwise} \end{cases}$$

$$\delta_{m\alpha'\alpha''k} = \begin{cases} 1, & \text{if box } \alpha' \text{ is on the } m^{th} \text{ side of box } \alpha'' \text{ in vehicle } k \\ 0, & \text{otherwise} \end{cases}$$

s_{ik}: time at which vehicle k begins serving customer i

$z'_{\alpha k}$, $z''_{\alpha k}$, $z'''_{\alpha k}$: box α's starting positions inside vehicle k for x, y and z coordinates, respectively $\alpha \in O$, $k \in K$, $z'_{\alpha k}$, $z''_{\alpha k}$, $z'''_{\alpha k} \in R^+$

Objective function:

$$\text{Min } f_1 \sum_{k=1}^{v} r_k y_k + \tag{1}$$

$$f_2 \sum_{k=1}^{v} \sum_{i=0}^{n} \sum_{j=1}^{n+1} c_{ijk} x_{ijk} \tag{2}$$

subject to:

$$\sum_{k=1}^{v}\sum_{j=1}^{n+1} x_{ijk} = 1, \qquad i = 1,\dots, n \qquad (3)$$

$$\sum_{j=1}^{n} x_{0jk} = y_k, \qquad k = 1,\dots,v \qquad (4)$$

$$\sum_{i=1}^{n} x_{i(n+1)k} = y_k, \qquad k = 1,\dots,v \qquad (5)$$

$$\sum_{i=0}^{n} x_{ilk} - \sum_{j=1}^{n+1} x_{ljk} = 0, \qquad l=1,\dots, n \quad k=1,\dots,v \qquad (6)$$

$$s_{i,k} + t_i + c_{ijk} - s_{j,k} \leqslant M_1 (1 - x_{i,j,k}), \qquad \langle i, j\rangle \in A \quad k=1,\dots v \qquad (7)$$

$$s_{ik} \geq earliest_i, \qquad i = 0, \dots, n+1 \quad k = 1, \dots, v \qquad (8)$$

$$s_{ik} \leq latest_i, \qquad i = 0, \dots, n+1 \quad k = 1, \dots, v \qquad (9)$$

$$\sum_{\alpha=1}^{o} q_\alpha \gamma_{\alpha k} \leq Q_k \qquad k=1,\dots,v \qquad (10)$$

$$\sum_{k=1}^{v} \gamma_{\alpha k} = 1, \qquad \alpha=1,\dots,o \qquad (11)$$

$$\sum_{j=1}^{n+1} x_{ijk} = \gamma_{\alpha k}, \qquad i=0,\dots, n \quad k=1,\dots, v \quad \forall_{\alpha \in e(i)} \qquad (12)$$

$$z'_{\alpha k} - L_k + l_\alpha \leq (1 - \gamma_{\alpha k}) M_2, \qquad \alpha = 1, \dots, o \quad k = 1, \dots, v \qquad (13)$$

$$z''_{\alpha k} - W_k + w_\alpha \leq (1 - \gamma_{\alpha k}) M_2, \qquad \alpha = 1, \dots, o \quad k = 1, \dots, v \qquad (14)$$

$$z'''_{\alpha k} - H_k + h_\alpha \leq (1 - \gamma_{\alpha k}) M_2, \qquad \alpha = 1, \dots, o \quad k = 1, \dots, v \qquad (15)$$

$$z'_{\alpha'' k} + l_{\alpha''} \leqslant z'_{\alpha' k} + (1 - \delta_{1\alpha'\alpha'' k}) M_2, \qquad \alpha', \alpha'' \in O \mid \alpha' < \alpha'' \quad k=1,\dots,v \qquad (16)$$

$$z'_{\alpha' k} + l_{\alpha'} \leq z'_{\alpha'' k} + (1 - \delta_{2\alpha'\alpha'' k}) M_2, \qquad \alpha', \alpha'' \in O \mid \alpha' < \alpha'' \quad k=1,\dots,v \qquad (17)$$

$$z''_{\alpha'' k} + w_{\alpha''} \leq z''_{\alpha' k} + (1 - \delta_{3\alpha'\alpha'' k}) M_2 \qquad \alpha', \alpha'' \in O \mid \alpha' < \alpha'' \quad k=1,\dots,v \qquad (18)$$

$$z''_{\alpha' k} + w_{\alpha'} \leq z''_{\alpha'' k} + (1 - \delta_{4\alpha'\alpha'' k}) M_2, \qquad \alpha', \alpha'' \in O \mid \alpha' < \alpha'' \quad k=1,\dots,v \qquad (19)$$

$$z'''_{\alpha'' k} + h_{\alpha''} \leq z'''_{\alpha' k} + (1 - \delta_{5\alpha'\alpha'' k}) M_2, \qquad \alpha', \alpha'' \in O \mid \alpha' < \alpha'' \quad k=1,\dots,v \qquad (20)$$

$$z'''_{\alpha' k} + h_{\alpha'} \leq z'''_{\alpha'' k} + (1 - \delta_{6\alpha'\alpha'' k}) M_2, \qquad \alpha', \alpha'' \in O \mid \alpha' < \alpha'' \quad k=1,\dots,v \qquad (21)$$

$$\sum_{m=1}^{6} \delta_{m\alpha'\alpha'' k} \geq 1 - (1 - \gamma_{\alpha' k}) - (1 - \gamma_{\alpha'' k}) \quad \alpha', \alpha'' \in O \mid \alpha' < \alpha'' \quad k=1,\dots,v \qquad (22)$$

$$\delta_{1\alpha'\alpha'' k} + \delta_{2\alpha'\alpha'' k} \leq 1 \qquad \alpha', \alpha'' \in O \mid \alpha' < \alpha'' \quad k=1,\dots,v \qquad (31)$$

$$\delta_{3\alpha'\alpha'' k} + \delta_{4\alpha'\alpha'' k} \leq 1 \qquad \alpha', \alpha'' \in O \mid \alpha' < \alpha'' \quad k=1,\dots,v \qquad (32)$$

$$\delta_{5\alpha'\alpha'' k} + \delta_{6\alpha'\alpha'' k} \leq 1 \qquad \alpha', \alpha'' \in O \mid \alpha' < \alpha'' \quad k=1,\dots,v \qquad (33)$$

$$\gamma_{\alpha' k} + \gamma_{\alpha'' k} \geq 2\delta_{m\alpha'\alpha'' k} \qquad \alpha', \alpha'' \in O \mid \alpha' < \alpha'' \quad k=1,\dots,v \quad m=1,\dots,6 \qquad (34)$$

$$x_{iik} = 0 \qquad i = 0,\dots, n+1 \quad k=1,\dots,v \qquad (35)$$

$x_{ijk} \in \{0, 1\}$	$\forall$ i, j, k	(36)
$y_k \in \{0, 1\}$	$\forall$ k	(37)
$s_{ik} \geq 0$	$\forall$ i, k	(38)
$\gamma_{\alpha k} \in \{0, 1\}$	$\forall$ α, k	(39)
$z'_{\alpha k} \geq 0$	$\forall$ α, k	(40)
$z''_{\alpha k} \geq 0$	$\forall$ α, k	(41)
$z'''_{\alpha k} \geq 0$	$\forall$ α, k	(42)
$\delta_{m\alpha'\alpha''k} \in \{0, 1\}$	$\forall$ m, α', α'', k	(43)

Objective function (1), (2) and constraints (3) to (15) are formulated as in Moura and Oliveira (2009). The number of vehicles (1) and total routing time (2) are minimized in the objective function with their weights f_1 and f_2, respectively. Note that, as in Moura and Oliveira (2009), loading of boxes is considered as a feasibility problem and therefore the objective function does not include any loading related criterion. Constraints (3) guarantee that only one vehicle is used from client *i* to client *j*. Constraints (4) ensure that if a vehicle is used then it departs from the depot. Constraints (5) make sure that if a vehicle is used, it returns back to the depot. Constraints (6) are flow conservation constraints; if a vehicle arrives to a node, then it has to leave from that node to another one. Constraints (7) guarantee if a vehicle visits client *j* after client *i*; then starting time of the service in client *j* is later than arriving of the vehicle to that city. Constraints (8) and (9) ensure time window of each client. Starting time of the service in client *i* should be greater than the earliest start time and less than the latest start time. Constraints (10) guarantee that the total weight of the boxes in each vehicle can not exceed the capacity of the vehicle. Constraints (11) make sure that each box is transported by only one vehicle. Constraints (12) ensure that all boxes of the same city must be transported by the same vehicle. Constraints (13)-(15), together with constraints (40)-(42), check for each dimension whether the box is appropriate or not for the vehicle.

Constraints (16)-(21) guarantee that each box pair, which is placed inside the same vehicle does not intersect at each axis. (front or back; left or right; above or below). Note that, constraints (16)-(21) are reformulation of the same numbered constraints in Moura and Oliveira (2009). If two boxes are not in the same vehicle then these constraints are redundant. Therefore, instead of $\delta_{m\alpha'\alpha''k}$ in Moura and Oliveira (2009), we use negation of the binary variables as (1- $\delta_{m\alpha'\alpha''k}$). Constraints (22) ensure that if any two boxes are assigned to the same vehicle, at least one of the positioning variables must be equal to 1. Different from Constraints (22) in Moura and Oliveira (2009), we enforce to take value for these relative positioning variables. Since any two boxes can be above or below; right or left; front or back of each other, we formulate constraints (31)-(33) to avoid assignment of two opposite directional binary variables to 1 simultaneously for each axis. Moreover, if any two boxes are not in the same vehicle, there is no need to compare relative positions of them. Hence, we formulate (34) as a valid inequality. Visiting the same client more than once is forbidden with constraints (35). Finally, domains of the variables are given in (36)-(43). The number of variables and constraints in this model are bounded by $6 \times |O| \times (|O|-1) \times |K| / 2$ and $((|O|^2-1)/2) \times |K|$.

4.1 A Numerical Example

In this section, we provide a numerical example to demonstrate the integration of vehicle routing problem with time windows (VRTWLP) and container loading problem (CLP) and application of the mixed integer programming (MIP) model given in the previous section. An instance is generated for demonstration includes 8 clients (|N|=8), 25 boxes (|O|=25), 2 vehicles (|K|=2) where corresponding parameters are provided in the following tables. Note that since there are 8 real clients to be visited, due to model formulation, we label depot as both the first (i=0) and the last client (i=9) to be visited in the following tables. Dimensions of the each boxes were generated arbitrarily for the 25 boxes. (Appendix-2)

Table 7 List of boxes of each client

Client (i)	Boxes
1	1, 2, 3
2	4, 5, 6, 7
3	8, 9, 10, 11
4	12, 13, 14
5	15, 16, 17, 18
6	19, 20
7	21, 22, 23
8	24, 25

Table 8 Attributes of each vehicle.

Vehicle (k)	Length (L_k)	Width (W_k)	Height (H_k)	Weight (Q_α)
1	8	3	4	50
2	5	2	3	50

Table 9 Service Time (t_{ik}) of each vehicle at each client

	Vehicle (k)	
Client (i)	1	2
0 (depot)	0	0
1	15	13
2	18	15
3	24	20
4	20	18
5	21	21
6	25	25
7	24	23
8	18	16
9 (depot)	0	0

Table 10 Time Window (S_{ik}) for serving each client

Client (i)	earliest_i	latest_i
0 (depot)	0	-
1	0	300
2	0	400
3	0	320
4	0	220
5	0	500
6	0	320
7	0	280
8	0	350
9 (depot)	0	-

In addition to above input data, we assume that fixed cost of vehicles (h_k) are 45 and 30 units for vehicle k=1 and k=2, respectively. Furthermore, we assume that the weights of sum of the traveling time and the vehicle usage costs (f_1 and f_2) are equally important.

Table 11 Traversing time for the first vehicle

c_{ij1}	0(depot)	1	2	3	4	5	6	7	8	9(depot)
0 (depot)	-	18	25	20	30	17	45	28	72	-
1	-	-	21	35	36	19	28	30	35	48
2	-	38	-	39	42	51	59	30	29	30
3	-	40	20	-	71	16	32	52	19	23
4	-	48	37	19	-	27	36	27	26	19
5	-	34	37	38	29	-	27	34	17	22
6	-	34	33	21	24	55	-	12	34	65
7	-	45	32	78	21	30	35	-	45	25
8	-	33	22	41	35	33	38	39	-	45
9 (depot)	-	-	-	-	-	-	-	-	-	-

Table 12 Traversing times for the second vehicle

c_{ij2}	0(depot)	1	2	3	4	5	6	7	8	9(depot)
0 (depot)	-	15	23	15	30	14	39	37	70	-
1	-	-	20	32	29	17	24	27	33	45
2	-	30	-	37	40	50	55	29	29	42
3	-	51	59	-	30	29	23	48	37	19
4	-	27	36	27	-	26	19	34	37	38
5	-	29	27	34	17	-	22	34	17	22
6	-	34	41	35	33	38	-	39	23	24
7	-	55	12	34	65	45	32	-	78	21
8	-	30	35	32	22	41	28	37	-	45
9 (depot)	-	-	-	-	-	-	-	-	-	-

The mathematical model is run for 2 hours on an AMD A6-3400M APU with 1.40 Ghz and 4 GB of RAM, running a Windows XP operative system and solved using IBM ILOG CPLEX 12.5 solver (IBM ILOG, 2015). Service start time of each vehicle at assigned cities in the feasible solution after 2 hours are given in the following tables which shows which verify feasibility of the solution in terms of vehicle routing with time windows constraints formulated in (3)-(10) and (35).

Table 13 Service start time at clients for vehicle 1

Cities (*i*)	1	2	7	4	3
S_{i1}	18	54	175	220	320

Table 14 Service start time at clients for vehicle 2

Cities (i)	5	8	6
S_{i2}	14	276	320

We use Rhinoceros 3D version 5 (Rhino3D, 2015) to visualize the solution of container loading problem. First, we show sides, directions and *x*, *y*, *z* coordinate labels assumed as below. (Fig. 10)

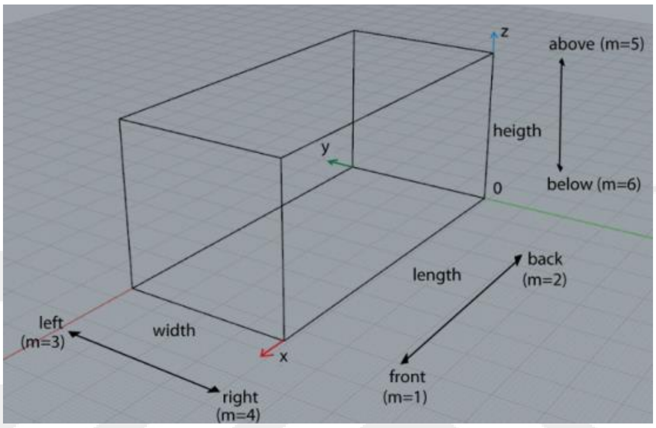
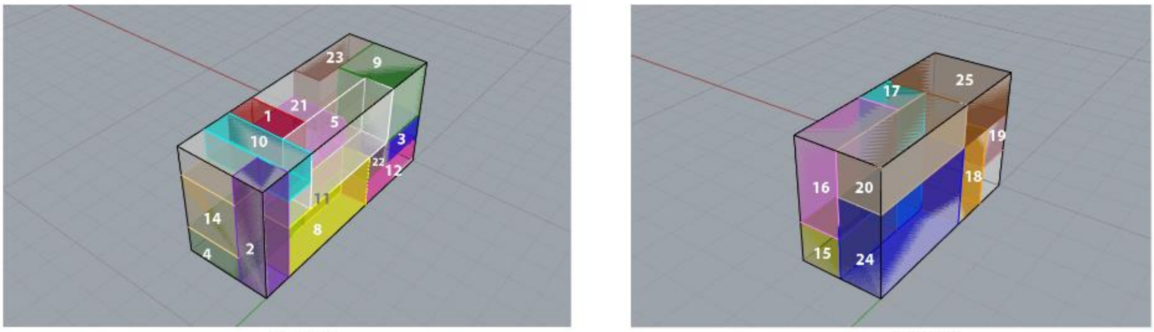


Figure 11 Sides $m \in Side$ of each vehicle

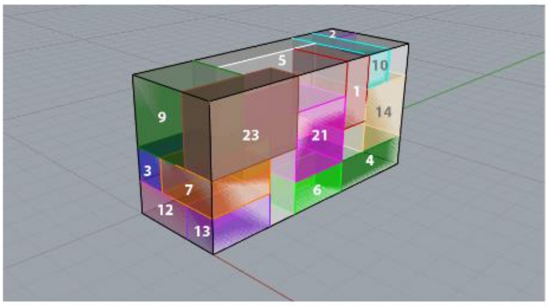
Assignment of boxes to each vehicle is shown in Figure 11.



Vehicle 1 Vehicle 2

Figure 12 Assignment of boxes to each vehicle, front view

Since some boxes assigned to vehicle 1 are not visible in front view, in the following figure, we give the back view.



Vehicle 1 Back View

Figure 13 Assignment of boxes to vehicle 1, back view

Since there are less number of boxes assigned to the second vehicle, we provide values of δ variables for each side, $m \in$ Side. Note that as mentioned in Moura and Oliveira (2009), relative placement has to be checked only once for each pair of boxes. In other words, any two boxes assigned to the same vehicle are compared while $\alpha' < \alpha''$. Therefore, there is no comparison again α'' and α' which means that $\delta_{m,\alpha'\alpha''k}$ values are 0 in Tables 15, 16 and 17 when $\alpha' < \alpha''$.

Table 15 Relative position of boxes in vehicle 2 on *y*-axis. a) If α' is in front of α'' then 1 otherwise 0. b) If $\alpha'$is at the back of α'' then 1, otherwise 0.

		α''							
		15	16	17	18	19	20	24	25
α'	15	-	0	1	1	1	0	0	0
	16	-	-	1	1	1	0	0	1
	17	-	-	-	1	1	0	0	1
	18	-	-	-	-	1	0	0	0
	19	-	-	-	-	-	0	0	0
	20	-	-	-	-	-	-	0	1
	24	-	-	-	-	-	-	-	0
	25	-	-	-	-	-	-	-	-

a.

		α''							
		15	16	17	18	19	20	24	25
α'	15	-	0	0	0	0	0	0	0
	16	-	-	0	0	0	0	0	0
	17	-	-	-	0	0	0	0	0
	18	-	-	-	-	0	0	1	0
	19	-	-	-	-	-	1	1	0
	20	-	-	-	-	-	-	0	0
	24	-	-	-	-	-	-	-	0
	25	-	-	-	-	-	-	-	-

b.

Table 16 Relative position of boxes in vehicle 2 on the *x*-axis. a) If α' is on left of α'' then 1 otherwise 0. b) If $\alpha'$is on the right of α'' then 1 otherwise 0.

		α''							
		15	16	17	18	19	20	24	25
α'	15	-	0	0	0	0	1	1	0
	16	-	-	0	0	0	1	1	0
	17	-	-	-	0	0	1	1	0
	18	-	-	-	-	0	0	0	0
	19	-	-	-	-	-	0	0	0
	20	-	-	-	-	-	-	0	0
	24	-	-	-	-	-	-	-	0
	25	-	-	-	-	-	-	-	-

a.

		α''							
		15	16	17	18	19	20	24	25
α'	15	-	0	0	0	0	0	0	0
	16	-	-	0	0	0	0	0	0
	17	-	-	-	0	0	0	0	0
	18	-	-	-	-	0	0	0	0
	19	-	-	-	-	-	0	0	0
	20	-	-	-	-	-	-	0	0
	24	-	-	-	-	-	-	-	0
	25	-	-	-	-	-	-	-	-

b.

Table 17 Relative position of boxes in vehicle 2 on the *z*-axis. a) If α' is above α'' then 1 otherwise 0. b) If $\alpha'$is below the α'' then 1 otherwise 0.

		α''							
		15	16	17	18	19	20	24	25
α'	15	-	0	0	0	0	0	0	0
	16	-	-	0	0	0	0	0	0
	17	-	-	-	0	0	0	0	0
	18	-	-	-	-	0	0	0	0
	19	-	-	-	-	-	0	0	0
	20	-	-	-	-	-	-	1	0
	24	-	-	-	-	-	-	-	0
	25	-	-	-	-	-	-	-	-

a.

		α''							
		15	16	17	18	19	20	24	25
α'	15	-	1	0	0	0	0	0	1
	16	-	-	0	0	0	0	0	0
	17	-	-	-	0	0	0	0	0
	18	-	-	-	-	0	1	0	1
	19	-	-	-	-	-	0	0	1
	20	-	-	-	-	-	-	0	0
	24	-	-	-	-	-	-	-	1
	25	-	-	-	-	-	-	-	-

b.

Positions inside vehicle 2 for *x*, *y* and *z* coordinates, respectively are given in the following table.

Table 18 Location of each box in vehicle 2

A	$z'_{\alpha 2}$	$z''_{\alpha 2}$	$z'''_{\alpha 2}$
15	3	1	0
16	3	1	1
17	2	1	0
18	1	0	0
19	0	0	1
20	2	0	2
24	2	0	0
25	0	0	2

Verification of the length, width and height limits of vehicle 2 is provided in the following table using positions and dimensions of each box.

Table 19 Verification of the length, width and height limits of vehicle 2

$z'_{15,2} + l_{15} \leq L_2$	which results in	$3 + 2 \leq 5$	✔
$z'_{16,2} + l_{16} \leq L_2$	which results in	$3 + 2 \leq 5$	✔
$z'_{20,2} + l_{18} \leq L_2$	which results in	$2 + 3 \leq 5$	✔
$z'_{24,2} + l_{19} \leq L_2$	which results in	$2 + 3 \leq 5$	✔
$z''_{16,2} + w_{16} \leq W_2$	which results in	$1+1 \leq 2$	✔
$z''_{25,2} + w_{25} \leq W_2$	which results in	$0+2 \leq 2$	✔
$z'''_{16,2} + h_{16} \leq H_2$	which results in	$1+2 \leq 3$	✔
$z'''_{17,2} + h_{17} \leq H_2$	which results in	$0+3 \leq 3$	✔
$z'''_{20,2} + h_{20} \leq H_2$	which results in	$2+1 \leq 3$	✔
$z'''_{25,2} + h_{25} \leq H_2$	which results in	$2+1 \leq 3$	✔

Finally, we demonstrate verification of disjunctive constraints (16) and (17) using boxes 15 (α'=15) and 17 (α''=17). Other constraints can be verified in a similar way. Constraints (16) and (17) can be reformulated for boxes 15 and 17 as in the following.

$$z'_{17,2} + l_{17} \leq z'_{15,2} + (1 - \delta_{1,15,17,2}) M_2 \quad (16)$$

$$z'_{15,2} + l_{15} \leq z'_{17,2} + (1 - \delta_{2,15,172}) M_2 \quad (17)$$

If numerical values of binary variables given in Table 18 and positioning variables are put in these inequalities, along with their respective lengths defined in Appendix 2 ($\delta_{1,15,17,2} = 1$, $\delta_{2,15,17,2} = 0$, $z'_{17,2}$=2, $z'_{15,2}$=3, l_{17}=1, l_{15}=2):

$$z'_{17,2} + l_{17} \leq z'_{15,2} + (1 - 1) M_2 \quad (16)$$

$$z'_{15,2} + l_{15} \leq z'_{17,2} + (1 - 0) M_2 \quad (17)$$

it obvious that constraint (17) will be redundant and constraint (16) makes $z'_{17,2} + l_{17} \leq z'_{15,}$ or with numerical values of $z'_{17,2}$ and l_{17} , $2 + 1 \leq 3$, which is a feasible solution.

4.2 Extended Mathematical Model with Extra Features

According to the 3-PL company, we need to include more constraints to improved mathematical model of Moura and Oliveira (2009) to satisfy their real need.

The current model does not include the following.

Capacity check of deci (dm^3): Like the weight capacity, for the white goods, summation of the deci is checked with the capacity limit of the each vehicle.

$d_\alpha = volume\ of\ box\ \alpha\ in\ deci$

$D_k = capacity\ of\ vehicle\ k\ in\ terms\ of\ deci$

$$\sum_{\alpha=1}^{O} d_\alpha \gamma_{\alpha k} \leq D_k \qquad k=1,\ldots,v$$

These constraints are used in the VRPTW part of the model. Before checking each dimension is appropriate for the vehicle dimension in the CLP part, it can cut the feasible region and reduces the feasible set.

The objective function includes both minimization of number of vehicles and total time travelled. With this new objective function with two different decision variables, we achieve the same goal with cost instead of time. Parameter h_k is the wage of a driver for vehicle k or the daily rent of vehicle k. As a new parameter, variable cost is included as a new parameter to calculate the fuel consumption. The unit measure of the objective function is cost per day.

$$\text{Min } \sum_{k=1}^{v} h_k y_k + \qquad (1)$$

$$\sum_{k=1}^{v} \sum_{i=0}^{n} \sum_{j=1}^{n+1} varcost_k c_{ijk} x_{ijk} \qquad (2)$$

CHAPTER 5: SOLUTION FOR EXTENDED MATHEMATICAL MODEL

In this chapter, the extended mathematical model will be formulated. As mentioned in the previous chapter, this model consists of the amended literature model and extra features of the 3-PL logistic company.

5.1 Data Generation

We use real data of the 3-PL logistics company. The distance between customers in each province are calculated by using Google Maps. Since this data is in km, it was converted into minutes by the average speed of each the vehicle (Appendix-3). After that, service time is same for all vehicles since the unloading operations are same. Time window used as the hours of operation in each store. Rental cost is valid for one month and fuel consumption is calculated by the formula in the Chapter 3 (Appendix-4). The dimensions of the each vehicle is provided by the 3-PL company (Appendix-5). The first data set includes 40 boxes. (Appendix-6) For the comparison with different data sets, 80 boxes (Appendix-7) and 120 boxes (Appendix-8) are obtained from the other daily data sets of 3-PL company.

5.2 Solution

Extended mathematical model was solved by using IBM ILOG CPLEX 12.5 solver. (Appendix-9) The model has 14860 decision variables and 33697 constraints. It continuous 22 hours until a memory error. Therefore, a gap limit or time limit need to specify to obtain a feasible solution.

After 17 hours of run, a feasible solution is obtained.

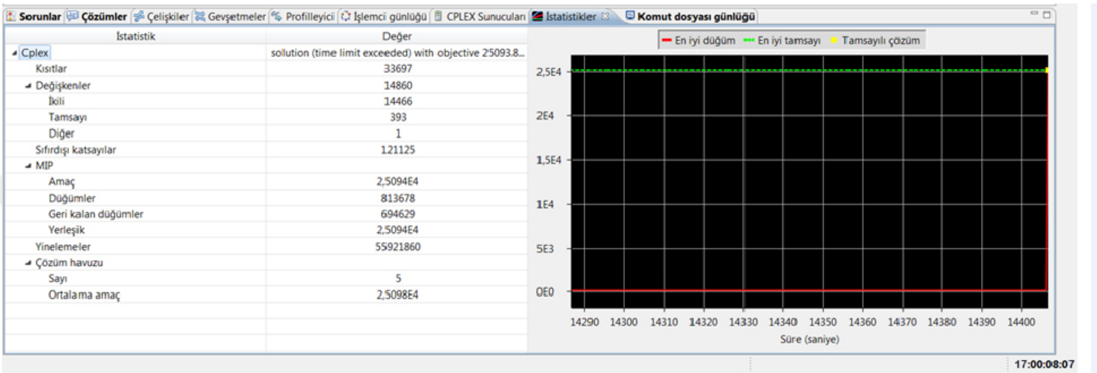


Figure 14 Extended OPL model, number of constraints and variables

Extended model includes three available vehicles for the problem. The feasible solution found utilizes only two vehicles with the objective value of 25.093,00 TRY. It includes both rental and fuel costs. The utilization of the vehicles are not so high. Each vehicle was used approximately half of its capacity. Utilization is calculated as below for each vehicle.

L_{box}: Length of the box

W_{box}: Width of the box

H_{box}: Height of the box

L_x: Length of vehicle *x*

W_x: Width of vehicle *x*

H_x: Height of vehicle *x*

$Boxes \in V_k$: box which loaded into vehicle k

Table 20 Calculation of Utilization

$$Utilization\ V_k = \frac{\sum_{boxes \in Vk}(L_{box}\ x\ W_{box}\ x\ H_{box})}{L_x\ x\ W_x\ x\ H_x}$$

The below figure give the placement assignments of each vehicle.

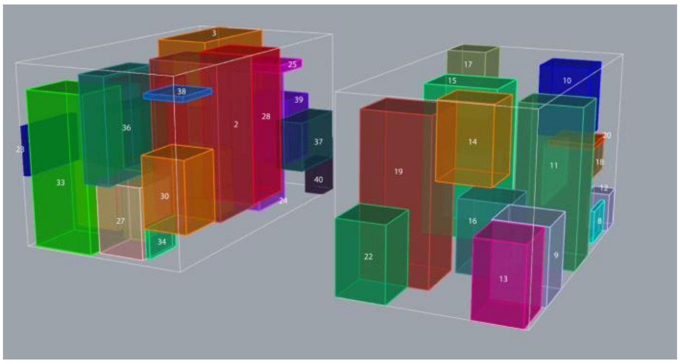


Figure 15 Extended Model- Container Loading Drawing of Vehicle 2 and 3

Two different vehicles shares the provinces rationally. Vehicle 2 transports to the north region of the Izmir and vehicle 3 transports the south region.

Vehicle 2: Kemalpasa- Özkanlar- Karşıyaka- Mavişehir- Aliağa- Ciğli- Kemalpaşa

Vehicle 3: Kemalpaşa- Bahçelievler- Balçova- Çeşme- Güzelyalı- Kemalpaşa

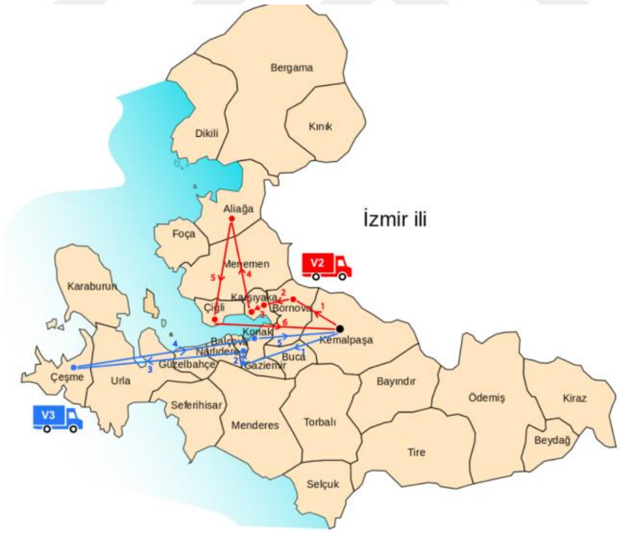


Figure 16 Extended Model-Vehicle Routing Drawing

Since the mathematical model was complex and hard to solve in a reasonable time, a heuristic approach applied on this mathematical model.

CHAPTER 6: A HEURISTIC FOR THE INTEGRATED PROBLEM

Both CVRPTW and CLP problems are NP-Hard problems. If the size of the problem increases, it becomes harder to obtain the optimal solution. According to the needs of the 3-PL company, the mathematical model is divided into two problems. Decomposition approach splits a large model into smaller models. It provides more natural modeling; coefficients tend to improve numerical stability, performance and correctness while avoiding a mix of large values (Arkalgud & Scott Rux, 2013).

6.1 Experimental Design

Now, we will discuss flow of Ecem Baris (EB) Heuristic are discussed.

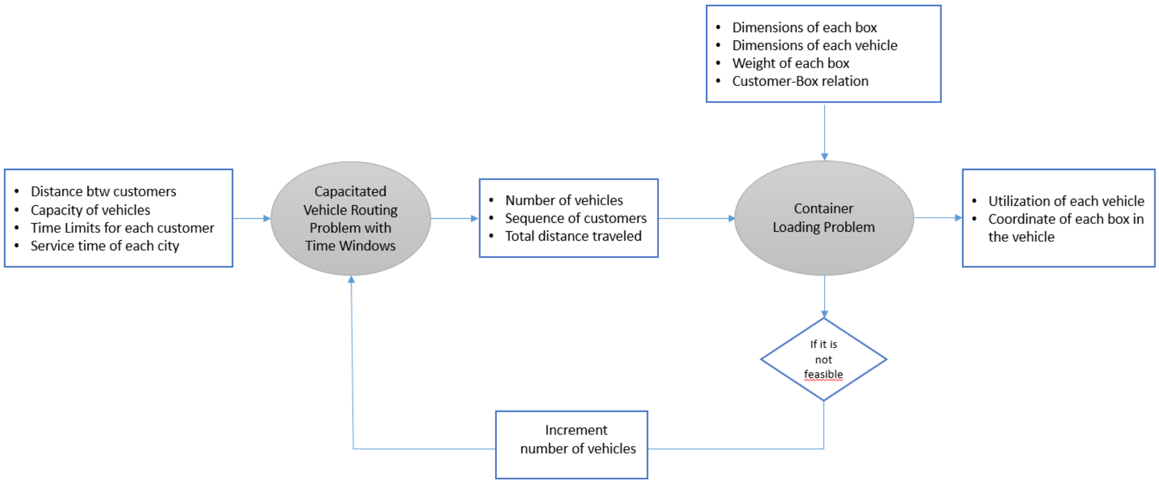


Figure 17 Flow chart for the EB Heuristic

First, inputs of the VRPTW are collected. These are the distance between customers, weight capacity of the vehicles, time limits of each customer and service time of each city. Afterwards, the number of required trucks, total distance traveled and sequence of cities are provided by this mathematical model. These outputs become inputs for the CLP. Moreover, dimension of each box, dimension of each vehicle, weight of each box and customer-box relations are added as new inputs for the CLP. Problem is solved in OPL, if the solution is not feasible with the additional constraints for capacity, then the model is run again with one additional truck. If the solution is feasible, then the problem results in the utilization of each vehicle, box coordination in the vehicle.

6.2 Computational Results

First part of the mathematical model (Vehicle Routing Problem) was solved in 9 seconds with an optimal value of 10.084,66 TRY with rental and fuel consumption cost.

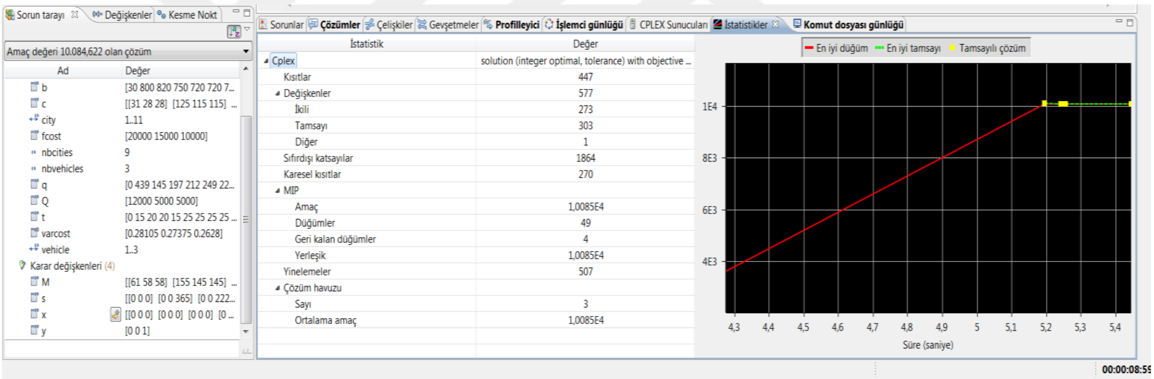


Figure 18 EB Heuristic- Vehicle Routing Number of constraints and variables

According to the flow chart, this model provide us how many trucks we need and the routing of the vehicles. These outputs will be our new inputs to our Container Loading Problem. The CLP part of the model was solved by OPL. (Appendix-10) This optimal solution uses only one vehicle and the route is

Vehicle 3: Kemalpaşa- Özkanlar-Bahçelievler- Güzelyalı- Balçova- Çeşme-
Karşıyaka- Mavişehir- Aliağa- Ciğli- Kemalpaşa

Figure 19 EB Heuristic-Vehicle Routing Drawing

Optimal solution includes only one vehicle, but we do not know whether it is enough or not for the container loading problem. Second part of the heuristic (Container Loading Problem) was updated to one vehicle instead of three. It decreases the size of the problem. Since we run the extended model for 17 hours, we ran this model for 16 hours, 59 minutes and 51 seconds. Since first composed model runs only 9 seconds.

Capacity of Vehicle 3 is enough and there is a feasible solution. Utilization is higher than the extended model.

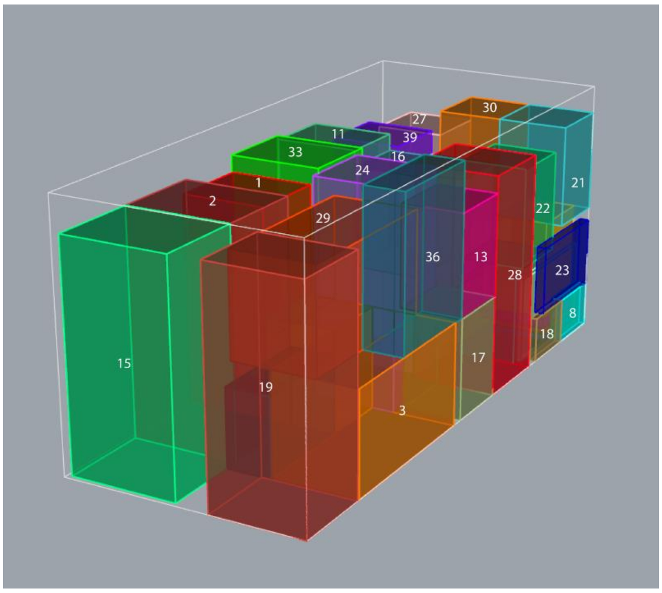


Figure 20 EB Heuristic-Container Loading Drawing of Vehicle 3

Performance of the EB Heuristic is tested over three different data sets; number of the boxes and vehicles are changed. To have a better comparison between extended model and heuristic model the below table was prepared. First data set only needs one vehicle to load, instead of two vehicles in the extended model. For the next two data sets, extended model does not provide a feasible solution in two hours. EB Heuristic is run with two vehicles and it does not give us a feasible solution, vehicle number has increased one by one. The first part of the EB heuristic was run until the second part was reached to a feasible solution. The total run time was calculated according to cumulative run hours until finding a feasible solution. Each run takes 2 hours. The feasible solution obtained by more vehicles, it results in a decrease in average utilization. If each run takes more than 10 hours, number of vehicles would be less, and the utilization would be more. To analyze the effect of the size of the problem set, two data set is obtained based on a 3-PL company, one with 80 boxes and the other one with 120 boxes.

Table 21 Comparison table for Extended Mathematical Model and EB Heuristic

	Data set 1			Data set 2		Data set 3	
	Extended Model	**EB (17 hours)**	**EB (2 hours)**	**Extended Model**	**EB**	**Extended Model**	**EB**
# Customers	9	9	9	9	9	9	9
# Vehicles	3	3	3	4	4	6	6
# Boxes	40	40	40	80	80	120	120
Gap	99%	85%	86%	-	78%	-	%80
Time Limit (given)	*17 hours*	*17 hours*	*2 hours*	*4 hours*	*4 hours*	*4 hours*	*4 hours*
Total Cost	25.093 TL	10.084 TL	10.084 TL	-	56.118 TL	-	77.150 TL
# of Vehicle Used	2	1	1	-	4	-	6
Total Distance Traveled (km)	357,4	334,9	334,9	-	421,5	-	568,2
Total Distance Traveled (min)	351	324	324	-	367	-	564
Weighted Average Utilization	0,40	0,74	0,74	-	0,42	-	0,46
Utilization V1 (23.716.000 cm3)	-	-	-	-	0,19	-	0,19
Utilization V2-(17.969.958 cm3)	0,31	-	-	-	0,63	-	0,94
Utilization V3-(21.244.150 cm3)	0,48	0,74	0,74	-	0,69	-	0,67
Utilization V4-(17.556.000 cm3)	-	-	-	-	0,19	-	0,29
Utilization V5-(18.963.000cm3)	-	-	-	-	-	-	0,34
Utilization V6-(17.776.000cm3)	-	-	-	-	-	-	0,39

CHAPTER 7: CONCLUSION AND FUTURE WORK

Green Logistics is the important topic in the real life problems and the transportation management is the core of logistic processes. In the literature, CLP and VRP is usually examined separately. In this study, we examine Vehicle Routing and Container Loading problems together. We firstly examine them separately. Their well-known features and types were defined. After that, combination of these two hard problems is analyzed, according to the needs of a 3-PL Company. We use real data and analyze the results. Afterwards, we extended the model. Although, an optimal solution of the mathematical model cannot be attained in a useful time, a decomposition heuristic is suggested for the integrated problem, in which the output of the VRTW problem becomes the input for the CLP. These two different approaches were compared. In the Chapter 6, the results were compared and it is obvious that the EB heuristic works better than the extended model.

This master thesis is about two combinatorial optimization problems. The mathematical model approach can be useful for the feasibility check, because feasible solutions can be obtained in a short time period. Moreover, both problems were solved in the literature separately using meta-heuristics. In addition to EB heuristic; meta-heuristics can be applied to this problem. Constructive heuristic methods can be another solution technique for this kind of NP-Hard problems to find an initial solution and then improved solutions can be obtained by searching the neighborhood.

Integration of these two problems is hard to solve with mathematical models. Since we decomposed the model into two; first part of the problem, VRPTW obtains optimal solution in a short time. The hardest part of the problem is the second part, because it includes a set of decision variables with four indices ($\delta_{m\alpha'\alpha''k}$). Instead of the extended model, EB heuristic gives us better solutions.

As a future study, the increase in the number of customers can be analyzed. In addition, for the CLP part of the EB Heuristic, the distances between boxes in the same vehicle can be added as a performance measure. According to the needs of logistics firms the additional constrains can be added. For the easy usage of the logistic firms, EB heuristic should be embedded in a system via a user interface.

Moreover, the deci capacity constraints can be checked if it is provide a cut or not in the EB heuristic. There are also another constraint set which are used to check each dimension of vehicle and box. Since the EB heuristic allows us to solve the problem in two parts; the first part can ensure a cut for the problem and they are actually redundant.

Another future work can be the landing of each box on the surface. Some of the boxes are positioned in the air in the solution of our model. To provide the usage in real life, the stability of the boxes has a great importance. The starting height of each box should be checked. The sum of the difference from ground to the floor of each box should also be minimized in the objective function.

REFERENCES

Arkalgud, N., & Scott Rux. (2013). *Examples of Flow Control in OPL.* Retrieved from Examples of Flow Control in OPL: http://www-01.ibm.com/support/docview.wss?uid=swg27039529&aid=1

Bell, J. E., & McMullen, P. R. (2004). Ant colony optimization techniques for the vehicle routing problem. *Advanced Engineering Informatics 1 (8)*, 41-48.

Bin, Y., Zhong-Zhen, Y., & Baozhen, Y. (2008). An improved ant colony optimization for vehicle routing problem. *European Journal of Operational Research*, 171-176.

Bischoff, E., & Ratcliff, M. (1995). Issues in the Development of Approaches to Container Loading. *Elsevier Science*, 377-390.

Bortfeldt, A., & Gehring, H. (1999). Zur Behandlung von Restriktionen bei der Stauraumoptimierung am Beispiel eines genetischen Algorithmus für das Containerbeladeproblem. In *Logistik Management* (pp. 83-100). Springer Berlin Heidelberg.

Bortfeldt, A., Mack, D., & Gehring, H. (2004). A parallel hybrid local search algorithm for the container loading problem. *Intl. Trans. in Op. Res*, 511-533.

Borthfeldt, A., & Wascher, G. (2012). *Container Loading Problems - State of the Art Review.* Magdeburg: Otto-von-Guericke-Universität .

Clarke, G., & Wright, J. W. (1964). Scheduling of Vehicles from a Central Depot to a Number of Delivery Points. *Operation Research*, 568-581.

Colorni, A., Maniezzo, M., & Trubian, M. (1994). Ant system for job- scheduling . *Jorbel-Belgian Journal of Operaion Research Statistics and Computer Science 34 (1)*, 39-53.

Desrochers, M., Lenstra, J.K., Savelsbergh, M.W.P., & Soumis, F. (1988). Vehicle routing with time windows: Optimization and approximation. In *Golden* (pp. 65–84). North-Holland, Amsterdam.

Doerner, K. F., Fuellerer, G., Gronalt, M., Hartl, R. F., & Iori, M. (2010). Metaheuristics for the vehicle routing problem with loading constraints. *European Journal of Operation Research*, 751-759.

Dorigo, M., Maniezzo, V., & Colorni, A. (1996). Ant system: optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Mans and Cybernatics 1 (26)*, 29-41.

Dyckhoff, H. (1990). A typology of cutting and packing problems. *European Journal of Operational Research, Vol. 44*, 145-159.

Eley, M. (2003, 25). A bottleneck assignment approach to the multiple container loading problem. *OR Spectrum*, 45-60.

Gehring, H., & Bortfeldt, A. (1997). A genetic algorithm for solving the container loading problem. *International Transactions in Operational Research,4* , 401-418.

Gendreau, M., Iori, M., Laporte, G., & Martello, S. (2006). A tabu search algorithm for a routing and container loading problem. *Informs*, 342-350.

Gürbüz, M., Akyokuş, S., Emiroğlu, İ., & Güran, A. (2009). *An Efficient Algorithm for 3D Rectangular Box Packing.* Skopje: Society for ETAI of Republic of Macedonia.

HAESSLER, R., & TALBOT, F. (1990). Load planning for shipments of low density products. *European Journal of Operational Research*, 289-299.

Hemminki, J., Leipala, T., & Nevalainen, O. (1998). On-line packing with boxes of different sizes. *International Journal of Production Research*, pp. 2225-2245.

Iori, M., & Martello, S. (2010). Routing problems with loading constraints. *Sociedad de Estadística e Investigación Operativa*, 4–27.

Junqueira, L., Oliveira, J. F., Caravilla, M. A., & Morabito, R. (2012). An optimization model for the vehicle routing problem withpractical three-dimensional loading constraints. *Intl. Trans. in Op. Res.*, 645–666.

Karayolları Genel Müdürlüğü Hız Sınırları. (n.d.). Retrieved 2016, from Karayolları Genel Müdürlüğü: http://www.kgm.gov.tr/Sayfalar/KGM/SiteTr/Trafik/HizSinirlari.aspx

Karayolları Trafik Yönetmeliği. (n.d.). Retrieved from Karayolları Genel Müdürlülüğü: http://www.kgm.gov.tr/SiteCollectionDocuments/KGMdocuments/Trafik/KanunYonetmelik/128MaddeDegisiklik.pdf

L., G., Taillard, E., & Dorrigo, M. (1997). *Ant Colonies for the QAP.* Lugano, Switzerland: Technical Report.

Leung, S. C., Zhang , Z., Zhang, D., Hua, X., & Lim, M. K. (2013). A meta-heuristic algorithm for heterogeneous fleet vehicle routing problems with two-dimensional loading constraints. *European Journal of Operational Research*, 199-210.

LIU, W.-Y., LIN, C.-C., & YU, C.-S. (2011). On the three-dimensional container packing prodblem under home delivery service. *Asia-Pacific Journal of Operational Research*, 1-20.

Martello, Silvano, Pisinger, D., & Paolo , T. (2000). *New trends in exact algorithms for the 0–1 knapsack problem.* European Journal of Operational Research.

Moghaddam, T., Rabbani, M., Shariat, M. A., & Safaei, N. (2006). *Solving a vehicle routing problem with soft time windows by a hybrid metaheuristic algorithm.* Tehran.

Moura, A. (2008). A Multi-Objective Genetic Algorithm for the Vehicle Routing with Time Windows and Loading Problem. In D. H. Bortfeldt, *Intelligent Decision Support* (pp. 187-201). Wiesbaden: Gabler.

Moura, A., & Oliveira, J. F. (2009). An integrated approach to the vehicle routing and container loading problems. *Springer*.

Osman, M. S., Ram, B., & Samanlioglu, F. (2014). Efficient approaches for furnace loading of cylindrical parts. *Elsevier*, 1548-1562.

Pisinger, D. (2002). Heuristics fır the Container Loading Problem. *European Journal of Operational Research*, 382-392.

Pollaris, H., Braekers, K., Caris, A., Jansenns, G. K., & Limbourg, S. (2014). Capacitated vehicle routing problem with sequence-based pallet loading and axle weight constraints. *EURO J Transp Logist, Springer*, DOI 10.1007/s13676-014-0064-2.

Republic of Turkey General Directorate of Highways. (n.d.). Retrieved from http://www.kgm.gov.tr

Robinson, J. A. (1980). A heuristic for packing boxes into a container. *Computer and Operations Research*, 147-156.

Scheithauer, G., & Terno, J. (1996). *A Heuristic Approach for Solving the Multi-Pallet Packing Problem.* Dresden: Dresden University of Technology.

Sixt, M. (1996). *Dreidimensionale Packprobleme; Lösungsverfahren basierend auf den Meta-Heuristiken.* Frankfurt, Germany: Peterlang.

Toth, P., & Vigo, D. (2002). *Vehicle Routing Problem.* Philadelphia: SIAM.

Wäscher, G., Haußner, H., & Schumann, H. (2007). An improved typology of cutting and packing problems. *European Journal of Operational Research*, 1109–1130.

APPENDICES

APPENDIX – 1. Mathematical Model for CVRPTW

```
int nbcities=...;
range city=1..nbcities+2; //depot=1&nb+2

int nbvehicles=...;
range vehicle=1..nbvehicles;


tuple arcs{
int i;
int j;
}
{arcs}arc={<i,j>|i in city,j in city:i!=j&&(i!=nbcities+2)&&(j!=1)&&(<i,j>!=<1,11>)}; //set of city pairs


execute{
for(a in arc){writeln(a)};
}

int c[arc,vehicle]=...; //travel time
int aa[city]=...; //lower time window
int b[city]=...; //upper time window
int q[city]=...; //total weight for city
int Q[vehicle]=...; //weight capacity
int t[city]=...; //service time of city

float varcost[vehicle]=...; //variable cost of vehicle k (fuel consumption per min)
dvar int+ M[arc,vehicle];
dvar int+ x[arc,vehicle] in 0..1;
dvar int+ s[city,vehicle];   //starting time
dvar int+ y[vehicle] in 0..1;
```

```
minimize sum(k in vehicle,a in arc:a.j>=2&&a.i<=nbcities+1)varcost[k]*c[a,k]*x[a,k]+sum(k in vehicle)y[k]*10000;

subject to{

forall(i in city:i<=nbcities+1&&i>=2){
      sum(k in vehicle,a in arc:(a.i==i))x[a,k]==1;   //3.constraint}

forall(a in arc,k in vehicle){M[a,k]==maxl(0,b[a.i]+t[a.i]+c[a,k]-aa[a.j]);}

forall(k in vehicle){
      sum(a in arc:a.i==1)x[a,k]==y[k];  //4.constraint}
forall(k in vehicle){
      sum(a in arc:a.j==nbcities+2)x[a,k]==y[k];  //5.constraint  }
forall(l in city,k in vehicle:l>=2&&l<=nbcities+1){
      sum(a in arc:a.j==l)x[a,k]-sum(a in arc:a.i==l)x[a,k]==0;  //6.constraint}

forall(a in arc,k in vehicle){
      s[a.i,k]+t[a.i]+c[a,k]-s[a.j,k]<=M[a,k]*(1-x[a,k]);  //7.constraint}

forall(i in city,k in vehicle){
      s[i,k]>=aa[i];  //8.constraint
      s[i,k]<=b[i];  //9.constraint}

forall(k in vehicle){
      sum(a in arc)q[a.i]*x[a,k]<=Q[k]*y[k];  //10.constraint  }

forall(k in vehicle,p in city){
      s[p,k]>=aa[p]+sum(a in arc:a.j==p)maxl(0,aa[a.j]-aa[a.i]+t[a.j]+c[a,k])*x[a,k];
      s[p,k]<=b[p]-sum(a in arc:a.i==p)maxl(0,b[a.i]-b[a.j]+t[a.i]+c[a,k])*x[a,k];}

forall(a in arc, i in city,j in city:((i==4)|| (i==5) || (j==4) || (j==5))){x[a,1]==0;} //1.araç büyük ve 4.,5. city
sokağına giremez.};

execute valuee {
```

```
for (var a in arc)
     for (var k=1;k<=nbvehicles;k++)
            writeln("x[",a,"][",k,"]=",x[a][k]);
            };
execute tme {
for (var i=1;i<=nbcities+2;i++)
     for (var k=1;k<=nbvehicles;k++)
            writeln("s[",i,"][",k,"]=",s[i][k]);
            };
execute arac {
     for (var k=1;k<=nbvehicles;k++)
            writeln("y[",k,"]=",y[k]);
            };
execute bigM {
for (var a in arc)
     for (var k=1;k<=nbvehicles;k++)
            writeln("M[",a,"][",k,"]=",M[a][k]);
            };
```

APPENDIX – 2. Attributes of each box for verification of the first model

Boxes(α)	Length (l_α)	Width (w_α)	Height (h_α)	Weight (q_α)
1	1	2	2	1
2	1	1	4	3
3	2	1	1	2
4	3	2	1	4
5	4	1	2	1
6	2	1	1	5
7	2	2	1	1
8	4	1	2	2
9	2	2	2	4
10	1	3	1	5
11	2	1	1	2
12	3	2	1	1
13	2	1	1	1
14	2	2	2	2
15	2	1	1	3
16	2	1	2	3
17	1	1	3	2
18	1	2	2	1
19	1	1	1	4
20	3	1	1	2
21	2	2	2	1
22	1	2	1	2
23	3	1	2	1
24	3	1	2	2
25	2	2	1	1

APPENDIX – 3. Data Generation – Google Maps distance (km)

		1	2	3	4	5	6	7	8	9	10	11
		Depo Kemalpaşa	MAVİŞEHİR	ÇEŞME	GÜZELYALI	BALÇOVA	BAHÇELİEVLER	ALİAĞA	KARŞIYAKA	ÇİĞLİ	ÖZKANLAR	Depo Kemalpaşa
1	Depo Kemalpaşa	1000	28,8	115	28,7	31,5	22,8	69,9	23,2	30,5	16,7	1000
2	MAVİŞEHİR	31,2	0	99,9	23,6	26,1	21,4	45,6	5,1	3,3	18,7	31,2
3	ÇEŞME	116	124	0	77,7	75,4	91,9	163	93,9	124	108	116
4	GÜZELYALI	28,9	22,1	79,1	0	3,4	7,5	64,3	17,6	25,2	14,9	28,9
5	BALÇOVA	33,4	26,6	74,1	4,5	0	13,4	91,8	22,1	29,7	19,5	33,4
6	BAHÇELİEVLER	28,9	21,9	81,5	4,6	7,7	0	64,1	17,3	23,4	14,7	28,9
7	ALİAĞA	72,8	45,4	162	65,2	89,2	63,1	0	50	45,4	58,1	72,8
8	KARŞIYAKA	26,9	5,2	95,6	19,3	22,2	17,2	51,4	0	8,8	12,7	26,9
9	ÇİĞLİ	22,4	3,4	121	24,6	48,6	22,2	47	8,4	0	17	22,4
10	ÖZKANLAR	21,6	15,6	108	17,6	20,6	15,5	56,8	10,9	15,8	0	21,6
11	Depo Kemalpaşa	1000	28,8	115	28,7	31,5	22,8	69,9	23,2	30,5	16,7	1000

APPENDIX – 4. Data Generation – Service Time, Time Windows, Rental Cost, Fuel Consumption

Service Time	Vehicle
depot	0
1	18
2	12
3	12
4	12
5	15
6	12
7	12
8	18
9	9
depot	0

Time Windows	a_i	b_i
depot	0	30
1	0	800
2	0	820
3	0	750
4	0	720
5	0	720
6	0	720
7	0	680
8	0	720
9	0	600
depot	0	960

Fixed Rent Cost (TRY)	
Vehicle 1	20000
Vehicle 2	15000
Vehicle 3	10000

Var Cost (TRY) per min	
Vehicle 1	0,28105
Vehicle 2	0,27375
Vehicle 3	0,2628

APPENDIX – 5. Dimension of the vehicles capacities

DIMENSION OF THE VEHICLES & CAPACITIES						
Vehicle	Lenght(cm)	Width	Height	Volume	Weight Capacity	Deci Capacity
1	490	220	220	23.716.000	7.200	7.905
2	387	213	218	17.969.958	2.500	5.990
3	482	205	215	21.244.150	2.500	7.081
4	380	210	220	17.556.000	2.500	5.852
5	420	210	215	18.963.000	2.500	6.321
6	400	202	220	17.776.000	2.500	5.925

APPENDIX – 6. Data Generation- Dimension of the white goods, α=40.

Retailer	Item	Product	Lenght (cm)	Width	Height	Weight (kg)	Desi
1	1	Refrigirator	70	72	173	99	291
1	2	Refrigirator	78	91	185	120	438
1	3	Deep-Freezer	150	72	85	109	306
1	4	Washing Machine	60	59	85	52	100
1	5	Air Conditioins-Int	72	25	53	13	32
1	6	Air Conditioins-Ext	92	38	68	46	79
2	7	TV	113	4	73	15	11
2	8	TV	53	5	38	14	3
2	9	Owen	60	60	85	65	102
2	10	Dishing Machine	60	60	85	51	102
3	11	Deep-Freezer	60	72	184	95	265
3	12	Microwave Owen	33	26	46	4	13
3	13	Washing Machine	60	53	85	47	90
3	14	Dishing Machine	60	60	85	51	102

4	15	Refrigirator	78	91	185	120	438
4	16	Dishing Machine	60	60	85	51	102
4	17	Wine Cooler	59	50	84	27	83
4	18	TV	53	5	38	14	3
5	19	Refrigirator	70	80	187	110	349
5	20	Cooktop	58	51	6	3	6
5	21	Owen	60	60	85	65	102
5	22	Washing Machine	60	56	85	48	95
5	23	TV	92	8	58	23	14
6	24	Refrigirator	70	72	183	108	307
6	25	Aspirator	60	60	10	5	12
6	26	Owen	60	60	85	65	102
6	27	Dishing Machine	60	60	85	51	102
7	28	Refrigirator	70	72	183	108	307
7	29	Deep-Freezer	150	72	85	109	306
7	30	Washing Machine	60	56	85	48	95
7	31	Dishing Machine	62	62	87	53	111
8	32	Water Dispenser	30	30	86	10	26
8	33	Refrigirator	70	80	187	110	349
8	34	TV	53	5	38	14	3
8	35	Washing Machine	60	50	85	45	85
8	36	Air Conditioins-Int	94	34	125	43	133
8	37	Air Conditioins-Ext	92	38	65	43	76
9	38	Cooktop	58	51	6	3	6
9	39	Dishing Machine	62	62	87	53	111
9	40	Microwave Owen	33	26	46	4	13

APPENDIX – 7. Data Generation- Dimension of the white goods, α=80.

Retailer	Item	Product	Lenght (cm)	Width	Height	Weight (kg)	Desi
1	1	Refrigirator	70	72	173	99	291
1	2	Refrigirator	78	91	185	120	438
1	3	Deep-Freezer	150	72	85	109	306
1	4	Washing Machine	60	59	85	52	100
1	5	Air Conditioins-Int	72	25	53	13	32
1	6	Air Conditioins-Ext	92	38	68	46	79
1	7	TV	113	4	73	15	11
1	8	TV	53	5	38	14	3
1	9	Owen	60	60	85	65	102
1	10	Dishing Machine	60	60	85	51	102
2	11	Deep-Freezer	60	72	184	95	265
2	12	Microwave Owen	33	26	46	4	13
2	13	Washing Machine	60	53	85	47	90
2	14	Dishing Machine	60	60	85	51	102
2	15	Refrigirator	78	91	185	120	438
3	16	Dishing Machine	60	60	85	51	102
3	17	Wine Cooler	59	50	84	27	83
3	18	TV	53	5	38	14	3
3	19	Refrigirator	70	80	187	110	349
3	20	Cooktop	58	51	6	3	6

3	21	Owen	60	60	85	65	102
3	22	Washing Machine	60	56	85	48	95
3	23	TV	92	8	58	23	14
3	24	Refrigirator	70	72	183	108	307
3	25	Aspirator	60	60	10	5	12
4	26	Owen	60	60	85	65	102
4	27	Dishing Machine	60	60	85	51	102
4	28	Refrigirator	70	72	183	108	307
4	29	Deep-Freezer	150	72	85	109	306
4	30	Washing Machine	60	56	85	48	95
4	31	Dishing Machine	62	62	87	53	111
4	32	Water Dispenser	30	30	86	10	26
4	33	Refrigirator	70	80	187	110	349
4	34	TV	53	5	38	14	3
4	35	Washing Machine	60	50	85	45	85
4	36	Air Conditioins-Int	94	34	125	43	133
4	37	Air Conditioins-Ext	92	38	65	43	76
5	38	Cooktop	58	51	6	3	6
5	39	Dishing Machine	62	62	87	53	111
5	40	Washing Machine	60	50	85	45	85
5	41	Air Conditioins-Int	94	34	125	43	133
5	42	Air Conditioins-Ext	92	38	65	43	76
5	43	Refrigirator	70	80	187	110	349
5	44	TV	53	5	38	14	3
5	45	Washing Machine	60	50	85	45	85
5	46	Refrigirator	70	80	187	110	349
6	47	Cooktop	58	51	6	3	6

6	48	Owen	60	60	85	65	102
6	49	Washing Machine	60	56	85	48	95
6	50	Washing Machine	60	50	85	45	85
6	51	Air Conditioins-Int	94	34	125	43	133
6	52	Air Conditioins-Ext	92	38	65	43	76
6	53	TV	53	5	38	14	3
7	54	Washing Machine	60	50	85	45	85
7	55	Refrigirator	70	80	187	110	349
7	56	Cooktop	58	51	6	3	6
7	57	Owen	60	60	85	65	102
7	58	Washing Machine	60	56	85	48	95
7	59	TV	92	8	58	23	14
7	60	TV	53	5	38	14	3
7	61	Washing Machine	60	50	85	45	85
7	62	Refrigirator	70	80	187	110	349
8	63	Cooktop	58	51	6	3	6
8	64	Owen	60	60	85	65	102
8	65	Refrigirator	78	91	185	120	438
8	66	Deep-Freezer	150	72	85	109	306
8	67	Washing Machine	60	59	85	52	100
8	68	Refrigirator	70	80	187	110	349
8	69	TV	53	5	38	14	3
8	70	Washing Machine	60	50	85	45	85
8	71	Refrigirator	78	91	185	120	438
9	72	Deep-Freezer	150	72	85	109	306
9	73	Refrigirator	78	91	185	120	438

9	74	Deep-Freezer	150	72	85	109	306
9	75	Washing Machine	60	59	85	52	100
9	76	TV	53	5	38	14	3
9	77	Washing Machine	60	50	85	45	85
9	78	Washing Machine	60	50	85	45	85
9	79	Air Conditioins-Int	94	34	125	43	133
9	80	Air Conditioins-Ext	92	38	65	43	76

APPENDIX – 8. Data Generation- Dimension of the white goods, α=120.

Retailer	Item	Product	Lenght (cm)	Width	Height	Weight (kg)	Deci
1	1	Refrigirator	70	72	173	99	291
1	2	Refrigirator	78	91	185	120	438
1	3	Deep-Freezer	150	72	85	109	306
1	4	Washing Machine	60	59	85	52	100
1	5	Air Conditioins-Int	72	25	53	13	32
1	6	Air Conditioins-Ext	92	38	68	46	79
1	7	TV	113	4	73	15	11
1	8	TV	53	5	38	14	3
1	9	Owen	60	60	85	65	102
1	10	Dishing Machine	60	60	85	51	102
1	11	Deep-Freezer	60	72	184	95	265
1	12	Microwave Owen	33	26	46	4	13
1	13	Washing Machine	60	53	85	47	90
1	14	Dishing Machine	60	60	85	51	102
1	15	Refrigirator	78	91	185	120	438
1	16	Dishing Machine	60	60	85	51	102
2	17	Wine Cooler	59	50	84	27	83
2	18	TV	53	5	38	14	3
2	19	Refrigirator	70	80	187	110	349
2	20	Cooktop	58	51	6	3	6

2	21	Owen	60	60	85	65	102
2	22	Washing Machine	60	56	85	48	95
2	23	TV	92	8	58	23	14
2	24	Refrigirator	70	72	183	108	307
2	25	Aspirator	60	60	10	5	12
2	26	Owen	60	60	85	65	102
2	27	Dishing Machine	60	60	85	51	102
2	28	Refrigirator	70	72	183	108	307
2	29	Deep-Freezer	150	72	85	109	306
2	30	Washing Machine	60	56	85	48	95
2	31	Dishing Machine	62	62	87	53	111
2	32	Water Dispenser	30	30	86	10	26
2	33	Refrigirator	70	80	187	110	349
2	34	TV	53	5	38	14	3
2	35	Washing Machine	60	50	85	45	85
2	36	Air Conditioins-Int	94	34	125	43	133
3	37	Air Conditioins-Ext	92	38	65	43	76
3	38	Cooktop	58	51	6	3	6
3	39	Dishing Machine	62	62	87	53	111
3	40	Washing Machine	60	50	85	45	85
3	41	Air Conditioins-Int	94	34	125	43	133
3	42	Air Conditioins-Ext	92	38	65	43	76
3	43	Refrigirator	70	80	187	110	349
3	44	TV	53	5	38	14	3
3	45	Washing Machine	60	50	85	45	85
3	46	Refrigirator	70	80	187	110	349
3	47	Cooktop	58	51	6	3	6
3	48	Owen	60	60	85	65	102
3	49	Washing Machine	60	56	85	48	95
3	50	Washing Machine	60	50	85	45	85
3	51	Air Conditioins-Int	94	34	125	43	133
4	52	Air Conditioins-Ext	92	38	65	43	76
4	53	TV	53	5	38	14	3
4	54	Washing Machine	60	50	85	45	85
4	55	Refrigirator	70	80	187	110	349
4	56	Cooktop	58	51	6	3	6
4	57	Owen	60	60	85	65	102
4	58	Washing Machine	60	56	85	48	95
4	59	TV	92	8	58	23	14

4	60	TV	53	5	38	14	3
4	61	Washing Machine	60	50	85	45	85
4	62	Refrigirator	70	80	187	110	349
4	63	Cooktop	58	51	6	3	6
4	64	Owen	60	60	85	65	102
4	65	Refrigirator	78	91	185	120	438
4	66	Deep-Freezer	150	72	85	109	306
4	67	Washing Machine	60	59	85	52	100
5	68	Refrigirator	70	80	187	110	349
5	69	TV	53	5	38	14	3
5	70	Washing Machine	60	50	85	45	85
5	71	Refrigirator	78	91	185	120	438
5	72	Deep-Freezer	150	72	85	109	306
5	73	Refrigirator	78	91	185	120	438
5	74	Deep-Freezer	150	72	85	109	306
5	75	Washing Machine	60	59	85	52	100
5	76	TV	53	5	38	14	3
5	77	Washing Machine	60	50	85	45	85
5	78	Washing Machine	60	50	85	45	85
5	79	Air Conditioins-Int	94	34	125	43	133
6	80	Air Conditioins-Ext	92	38	65	43	76
6	81	Wine Cooler	59	50	84	27	83
6	82	TV	53	5	38	14	3
6	83	Refrigirator	70	80	187	110	349
6	84	Washing Machine	60	50	85	45	85
6	85	Washing Machine	60	50	85	45	85
6	86	Refrigirator	70	80	187	110	349
6	87	Cooktop	58	51	6	3	6
6	88	Washing Machine	60	50	85	45	85
6	89	Washing Machine	60	50	85	45	85
6	90	Air Conditioins-Int	94	34	125	43	133
6	91	Air Conditioins-Ext	92	38	65	43	76
6	92	Refrigirator	70	72	173	99	291
6	93	Refrigirator	78	91	185	120	438
7	94	Deep-Freezer	150	72	85	109	306
7	95	TV	92	8	58	23	14
7	96	TV	53	5	38	14	3
7	97	Washing Machine	60	50	85	45	85
7	98	Refrigirator	70	80	187	110	349

7	99	Cooktop	58	51	6	3	6
7	100	Washing Machine	60	50	85	45	85
7	101	Washing Machine	60	50	85	45	85
7	102	Air Conditioins-Int	94	34	125	43	133
7	103	Air Conditioins-Ext	92	38	65	43	76
7	104	Wine Cooler	59	50	84	27	83
8	105	TV	53	5	38	14	3
8	106	Refrigirator	70	80	187	110	349
8	107	Refrigirator	70	72	173	99	291
8	108	Refrigirator	78	91	185	120	438
8	109	Deep-Freezer	150	72	85	109	306
8	110	Refrigirator	70	80	187	110	349
8	111	Cooktop	58	51	6	3	6
8	112	Washing Machine	60	50	85	45	85
8	113	Washing Machine	60	50	85	45	85
9	114	Air Conditioins-Int	94	34	125	43	133
9	115	Air Conditioins-Ext	92	38	65	43	76
9	116	Wine Cooler	59	50	84	27	83
9	117	TV	53	5	38	14	3
9	118	Refrigirator	70	72	173	99	291
9	119	Refrigirator	78	91	185	120	438
9	120	Deep-Freezer	150	72	85	109	306

APPENDIX – 9. Extended Model- Mathematical Model

```
int nbcities=...;
range city=1..nbcities+2; //depot=1&nb+2

int nbvehicles=...;
range vehicle=1..nbvehicles;

int nbboxes=...; //total number of boxes
range box=1..nbboxes;

int nbdirections=...;
```

```
range direction=1..nbdirections;

{int} custbox[city]=...;


tuple arcs{
int i;
int j;
}
{arcs}arc={<i,j>|i in city,j in city:i!=j&&(i!=nbcities+2)&&(j!=1)&&(<i,j>!=<1,11>)}; //set of city pairs


execute{
for(a in arc){writeln(a)};
}


int c[arc,vehicle]=...; //travel time
int a[city]=...; //lower time window
int b[city]=...; //upper time window
int l[box]=...; //length of box
int w[box]=...; //width of box
int h[box]=...; //height of box
int q[box]=...; //weight of box
int L[vehicle]=...; //length of vehicle
int W[vehicle]=...; //width of vehicle
int H[vehicle]=...; //height of vehicle
int Q[vehicle]=...; //weight of vehicle
int t[city,vehicle]=...; //service time of city i by vehicle k
int fcost[vehicle]=...;  //fixed cost of vehicle k (rent)
float varcost[vehicle]=...; //variable cost of vehicle k (fuel consumption per min)
int type[box]=...;
int M1=...;
int M2=...;

dvar int+ x[city,city,vehicle] in 0..1;
```

```
dvar int+ y[vehicle] in 0..1;
dvar int+ s[city,vehicle];   //starting time
dvar int+ gama[box,vehicle] in 0..1;
dvar int+ zl[box,vehicle]; //box coordination length
dvar int+ zw[box,vehicle]; //box coordination width
dvar int+ zh[box,vehicle]; //box coordination height
dvar int+ delta[direction,box,box,vehicle] in 0..1;

minimize sum(k in vehicle)fcost[k]*y[k]+sum(k in vehicle,a in
arc:a.j>=2&&a.i<=nbcities+1)varcost[k]*c[a,k]*x[a.i,a.j,k];

subject to{
forall(i in city:i<=nbcities+1&&i>=2){
      sum(k in vehicle,j in city:j>=2)x[i,j,k]==1;   //3.constraint
}
forall(k in vehicle){
      sum(j in city:j<=nbcities+1&&j>=2)x[1,j,k]==y[k];  //4.constraint
}
forall(k in vehicle){
      sum(i in city:i<=nbcities+1&&i>=2)x[i,nbcities+2,k]==y[k];  //5.constraint
}
forall(l in city,k in vehicle:l>=2&&l<=nbcities+1){
      sum(i in city:i<=nbcities+1)x[i,l,k]-sum(j in city:j>=2)x[l,j,k]==0;  //6.constraint
}

forall(a in arc,k in vehicle){
      s[a.i,k]+t[a.i,k]+c[a,k]-s[a.j,k]<=M1*(1-x[a.i,a.j,k]);  //7.constraint
}

forall(i in city,k in vehicle){
      s[i,k]>=a[i];  //8.constraint
      s[i,k]<=b[i];  //9.constraint
}
forall(k in vehicle){
      sum(alfa in box)q[alfa]*gama[alfa,k]<=Q[k];  //10.constraint
}
```

```
forall(alfa in box){
      sum(k in vehicle)gama[alfa,k]==1;  //11.constraint
}

forall(i in city,k in vehicle,alfa in box:i<=nbcities+1&&alfa in custbox[i]){
ll:   sum(j in city:j>=2)x[i,j,k]<=gama[alfa,k];  //12.constraint
}

forall(alfa in box,k in vehicle){
      zl[alfa,k]-L[k]+l[alfa]<=(1-gama[alfa,k])*M2;  //13.constraint
      zw[alfa,k]-W[k]+w[alfa]<=(1-gama[alfa,k])*M2;  //14.constraint
      zh[alfa,k]-H[k]+h[alfa]<=(1-gama[alfa,k])*M2;  //15.constraint
}
forall(alfa1 in box,alfa2 in box,k in vehicle:alfa1<=nbboxes-1&&alfa2>=alfa1+1){
      zl[alfa2,k]+l[alfa2]<=zl[alfa1,k]+(1-delta[1,alfa1,alfa2,k])*M2; //16.constraint
      zl[alfa1,k]+l[alfa1]<=zl[alfa2,k]+(1-delta[2,alfa1,alfa2,k])*M2; //17.constraint
      zw[alfa2,k]+w[alfa2]<=zw[alfa1,k]+(1-delta[3,alfa1,alfa2,k])*M2; //18.constraint
      zw[alfa1,k]+w[alfa1]<=zw[alfa2,k]+(1-delta[4,alfa1,alfa2,k])*M2;} //19.constraint

forall(alfa1 in box,alfa2 in box:alfa1<=nbboxes-1&&alfa2>=alfa1+1)sum(m in direction,k in
vehicle:m<=2)delta[m,alfa1,alfa2,k]<=1; //22.constraint
forall(alfa1 in box,alfa2 in box:alfa1<=nbboxes-1&&alfa2>=alfa1+1)sum(m in direction,k in
vehicle:m>2&&m<=4)delta[m,alfa1,alfa2,k]<=1; //23.constraint
forall(alfa1 in box,alfa2 in box:alfa1<=nbboxes-1&&alfa2>=alfa1+1)sum(m in direction,k in
vehicle:m>4&&m<=6)delta[m,alfa1,alfa2,k]<=1; //24.constraint

forall(alfa1 in box,alfa2 in box,k in vehicle:alfa1<=nbboxes-
1&&alfa2>=alfa1+1)delta[1,alfa1,alfa2,k]+delta[2,alfa1,alfa2,k]+delta[3,alfa1,alfa2,k]+delta[4,alfa1,alfa2,k]+delta[
5,alfa1,alfa2,k]+delta[6,alfa1,alfa2,k]>=1-(1-gama[alfa1,k])-(1-gama[alfa2,k]);//25.constraint
forall(alfa1 in box,alfa2 in box,k in vehicle,m in direction:alfa1<=nbboxes-
1&&alfa2>=alfa1+1)gama[alfa1,k]+gama[alfa2,k]>=2*delta[m,alfa1,alfa2,k];//26.constraint
forall(k in vehicle,i in city){x[i,i,k]==0; }//27.constraint


};
```

APPENDIX – 10 Integrated Mathematical Model- CLP Part

```
int nbcities=...;
range city=1..nbcities+2; //depot=1&nb+2

int nbvehicles=...;
range vehicle=1..nbvehicles;

int nbboxes=...; //total number of boxes
range box=1..nbboxes;

int nbdirections=...;
range direction=1..nbdirections;

{int} custbox[city]=...;


tuple arcs{
int i;
int j;
}
{arcs}arc={<i,j>|i in city,j in city:i!=j&&(i!=nbcities+2)&&(j!=1)&&(<i,j>!=<1,11>)}; //set of city pairs


execute{
for(a in arc){writeln(a)};
}


int l[box]=...; //length of box
```

```
int w[box]=...; //width of box
int h[box]=...; //height of box
int q[box]=...; //weight of box
int L[vehicle]=...; //length of vehicle
int W[vehicle]=...; //width of vehicle
int H[vehicle]=...; //height of vehicle
int Q[vehicle]=...; //weight of vehicle

int M1=...;
int M2=...;
//int x[arc,vehicle]=...;
int y[vehicle]=...;
int gama[box,vehicle]=...;
dvar int+ zl[box,vehicle]; //box coordination length
dvar int+ zw[box,vehicle]; //box coordination width
dvar int+ zh[box,vehicle]; //box coordination height
dvar int+ delta[direction,box,box,vehicle] in 0..1;

minimize sum(alfa in box,k in vehicle) (zl[alfa,k]+zw[alfa,k]+zh[alfa,k]);


subject to{

forall(k in vehicle){
      sum(alfa in box)q[alfa]*gama[alfa,k]<=Q[k];  //10.constraint
}

forall(alfa in box){
      sum(k in vehicle)gama[alfa,k]==1;  //11.constraint
}

forall(alfa in box,k in vehicle){
      zl[alfa,k]-L[k]+l[alfa]<=(1-gama[alfa,k])*M2;  //13.constraint
      zw[alfa,k]-W[k]+w[alfa]<=(1-gama[alfa,k])*M2;  //14.constraint
      zh[alfa,k]-H[k]+h[alfa]<=(1-gama[alfa,k])*M2;  //15.constraint
}
```

```
forall(alfa1 in box,alfa2 in box,k in vehicle:alfa1<=nbboxes-1&&alfa2>=alfa1+1){
      zl[alfa2,k]+l[alfa2]<=zl[alfa1,k]+(1-delta[1,alfa1,alfa2,k])*M2; //16.constraint
      zl[alfa1,k]+l[alfa1]<=zl[alfa2,k]+(1-delta[2,alfa1,alfa2,k])*M2; //17.constraint
      zw[alfa2,k]+w[alfa2]<=zw[alfa1,k]+(1-delta[3,alfa1,alfa2,k])*M2; //18.constraint
      zw[alfa1,k]+w[alfa1]<=zw[alfa2,k]+(1-delta[4,alfa1,alfa2,k])*M2; //19.constraint
      zh[alfa2,k]+h[alfa2]<=zh[alfa1,k]+(1-delta[5,alfa1,alfa2,k])*M2; //20.constraint
      zh[alfa1,k]+h[alfa1]<=zh[alfa2,k]+(1-delta[6,alfa1,alfa2,k])*M2; //21.constraint
}

forall(alfa1 in box,alfa2 in box:alfa1<=nbboxes-1&&alfa2>=alfa1+1)sum(m in direction,k in
vehicle:m<=2)delta[m,alfa1,alfa2,k]<=1; //22.constraint
forall(alfa1 in box,alfa2 in box:alfa1<=nbboxes-1&&alfa2>=alfa1+1)sum(m in direction,k in
vehicle:m>2&&m<=4)delta[m,alfa1,alfa2,k]<=1; //23.constraint
forall(alfa1 in box,alfa2 in box:alfa1<=nbboxes-1&&alfa2>=alfa1+1)sum(m in direction,k in
vehicle:m>4&&m<=6)delta[m,alfa1,alfa2,k]<=1; //24.constraint

forall(alfa1 in box,alfa2 in box,k in vehicle:alfa1<=nbboxes-
1&&alfa2>=alfa1+1)delta[1,alfa1,alfa2,k]+delta[2,alfa1,alfa2,k]+delta[3,alfa1,alfa2,k]+delta[4,alfa1,alfa2,k]+delta[
5,alfa1,alfa2,k]+delta[6,alfa1,alfa2,k]>=1-(1-gama[alfa1,k])-(1-gama[alfa2,k]);//25.constraint
forall(alfa1 in box,alfa2 in box,k in vehicle,m in direction:alfa1<=nbboxes-
1&&alfa2>=alfa1+1)gama[alfa1,k]+gama[alfa2,k]>=2*delta[m,alfa1,alfa2,k];//26.constraint

};
```