Optimization Models for Inland Haulage Transportation of Containers

A thesis submitted for the degree of

Doctor of Philosophy

Department of Mathematical Sciences
University of Essex

by

Hajem Ati Daham

February, 2019

Dedicated to

The memories of my parents and my brothers Jasim and Ali.

My wife for her patience, kindness, help and support.

Acknowledgements

I would like to express my sincere gratitude to Almighty Allah, for giving me strength, courage and the best blessing to complete my Ph.D study and guide me all the way in my life. This thesis is the culmination of input, work and encouragement of many people who have helped and accompanied me for the years that I have spent at University of Essex. First of all, my deepest gratitude goes to all my family, particularly my died parents and brothers for supporting me spiritually throughout writing this thesis and my life in general. A huge debt of gratitude is extended to my dear wife Nadheema for her patient and for almost unbelievable support, she is the most important person in my world and I dedicate this thesis to her. You were always with me on rain days and at sleepless nights. You smiled when I smile and you provided the shoulder when I cried. You kept me going and you made this real. You are my pulse, my blood, my love...other best.

I would like to express my sincerest thanks to my supervisor, Dr. Xinan Yang , for her patience, motivation, and immense knowledge. Her guidance helped me in all the time of research and writing of this thesis. I could not have imagined having a better advisor and mentor for my Ph.D study. I feel extremely fortunate to work with her. Many thanks for her insightful comments and encouragement and for the important notes and information which incentive me to widen my research from various perspectives. Without her precious support it would not be possible to conduct this research. I learned a lot from her and I am very grateful to have her as my advisor.

For financial support, many many thanks go to the Iraqi Ministry of Higher Education and Scientific Research for sponsoring my scholarship.

Specially thanks to all Department of Mathematical Sciences staff for their help and Essex University for their services.

Finally, I would like to thank all my relatives and friends who pray and support me during my study.

Declaration

The work in this thesis is based on research carried out in the Department of Mathematical Sciences, University of Essex, United Kingdom. No part of this thesis has been submitted elsewhere for any other degree or qualification, and it is all my own work, unless referenced, to the contrary, in the text.

Abstract

In container intermodal transportation, a significant portion of the total cost arises from the inland transportation of containers. There are many parties (shipping lines, haulage companies, customers) sharing this operation as well as many restrictions that increase the complexity of this problem and make it NP-hard. Shipping lines and haulage companies tend to apply efficient optimization techniques to manage this process in a way to reduce the overall cost and to ensure that customers are satisfied. In this thesis, we focus on container inland transportation from the perspective of delivering 20ft and/or 40ft containers on a heterogenous fleet of trucks, between a single port and a list of customer locations and inland depots. We investigate three types of inland transportation problems: Homogenous Container Sizes, Heterogenous Container Sizes and Stripe and Discharge of Containers.

Each of the above problems has its own complexity but all have been classified as NP-hard problems. For this reason we will study these problems separately and the main contributions are describing, modelling, solving and analysing of the:

- Homogenous Container Sizes: an efficient assignment Mixed Integer Linear Programming (MILP) model is formulated which solves large scale instances in a reasonable solution time and can be implemented on variants of the container drayage problem.

- Heterogenous Container Sizes: a Mixed Integer Linear Programming (MILP) model for combining 20ft and 40ft, Stripe orders is designed, which solves more efficiently than its previous analog. For realistic instances, a decomposition and aggregation heuristic is designed and tested to be cost saving.

- Strip and Discharge of Containers: a Genetic Algorithm (GA) approach is designed and tested for solving large scale problems within a quick computational time and the result shows that combining the *Strip* and *Discharge* types with the usage of inland empty depots is cost and fleet saving.

Keywords: Inland Transportation; Combining Orders; An Assignment MILP; Heuristic Decomposition; Genetic Algorithm

Contents

List of Figures

List of Tables

List of Acronyms

ISO	International Standard Organisation
TEUs	Twenty-foot Equivalent Units
PCDT	Pairing of Containers in Drayage Transportation
MILP	Mixed Integer Linear Programming
VRP-SPDTW	Vehicle Routing Problem with Simultaneous Pickups and Deliveries and Time Windows
M-TSPTW	Multi-Travelling Salesman Problem with Time Windows
ILP	Integer Linear Programming
VNS	Variable Neighbourhood Search
CDPTW	Container Drayage Problems with Time Windows
LP	Linear Programming
CG	Column Generation
RMP	Restricted Master Problem
GA	Genetic Algorithm

Chapter 1

Introduction

1.1 Containerization in Logistics: An Overview

In the period before the 1950s, freight transportation had experienced many problems, such as damage to cargo and slow transportation [66]. This situation changed later, when the use of containers began and became a common method for global trade transportation. Containers as shown in (Figure 1.1) are boxes made from solid material which are characterized as secure and safe means for the transportation of goods and products for long distances and times. Containers form the most integral part of the entire shipping industry, and are created to store various kinds of products that need to be transported between different regions and areas. Moving by containers will protect contents during the long journeys they make and ensure that they are collected in one piece. The International Standard Organisation (ISO) classifies containers into standard sizes: 20ft, 40ft, 45ft, 48ft and 53ft, in which 20ft and 40ft are the most commonly used. Consequently, most trucks and vehicles are designed to carry these two types [71]. As such, depending on the type of products to be shipped or the special services needed, container units may vary in dimensions. Various types of shipping containers are being used today to meet requirements of all kinds of cargo shipping. For instance, the transportation of chilled and frozen products requires a special type of container called reefers containers which are provided with temperature regulators and have a carefully controlled low temperature, while dry storage containers are used for shipping of dry materials. Furthermore, tanks storage containers are used for transportation of liquid materials.

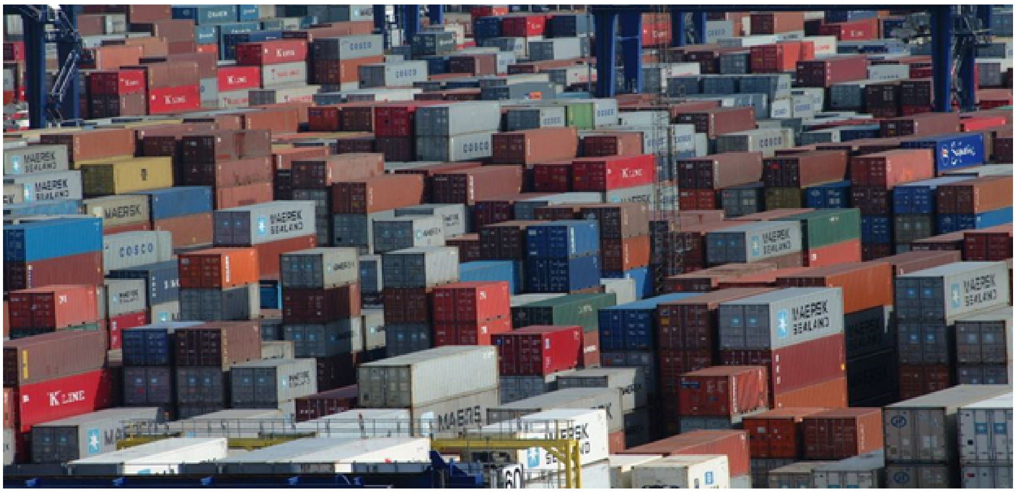

Figure 1.1: *Containers (A container terminal at the Port of Felixstowe)*

Rodrigue et al. [84] reported that the world traffic of containers, which is the number of containers being transported, has grown significantly with an average annual growth of 9.5%. However, containers throughput, which refers to the number of containers handled at ports, has an average annual growth of about 10.5% during the same period. This diverging emphasizes that the global trade represented by supply chains becomes more complex. The rising of both container traffic and throughput is related to the growth of international trade and the adoption of containers as an important method for both maritime shipping and inland transportation. Until 2009 the growth of containers throughput had increased continuously. However, the financial crisis of 2009-2010 resulted in a drop of about 49 million TEUs (9.3%) during that period. This was the first time that the global container flow level came down. In 2015, the growth of container throughput continues to increase to become about 687 million TEUs (see Figure 1.2).

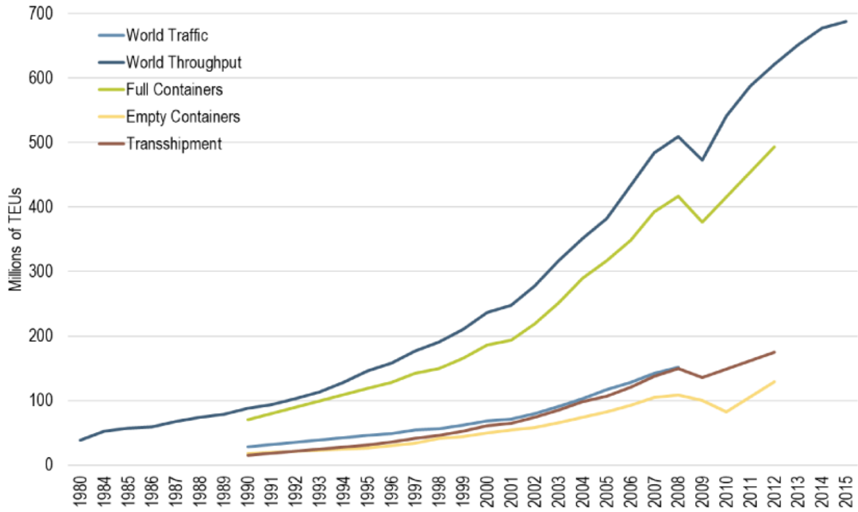

Figure 1.2: *Growth of Global containerized trade between 1980 and 2015. Source: Drewry Shipping Consultants and own elaboration.*

1.2 Intermodal Freight Transportation

Intermodal freight transportation, which is also called Multimodal transportation, is the transportation of goods and products as a containerized cargo from their origins to destinations by a combination of transportation modes [20] including trucks, rails and vessels (Figure 1.3). As shown in Figure 1.3, the procedure of this operation usually starts by the delivery of loaded containers from shippers directly to the port/terminal or to a rail yard by truck and then to the terminal. A vessel or barge will then move the loaded containers from the origin port to another destination port, from where the loaded containers will be delivered to the consignees by trucks or trains again. Combining different transportation modes is vital in the freight transportation industry. Many important factors can affect the decision of choosing the appropriate transportation modes, for instance, the overall cost, the service circumstances, the delivery speed and facilities for handling and packing at customers' locations. Each transportation mode has its own features and advantages, for instance, the benefit of using the truck (lorry) is the flexible and immediate delivery of cargos, while rail transportation allows the delivery of a larger number of cargos with less cost and less pollution.

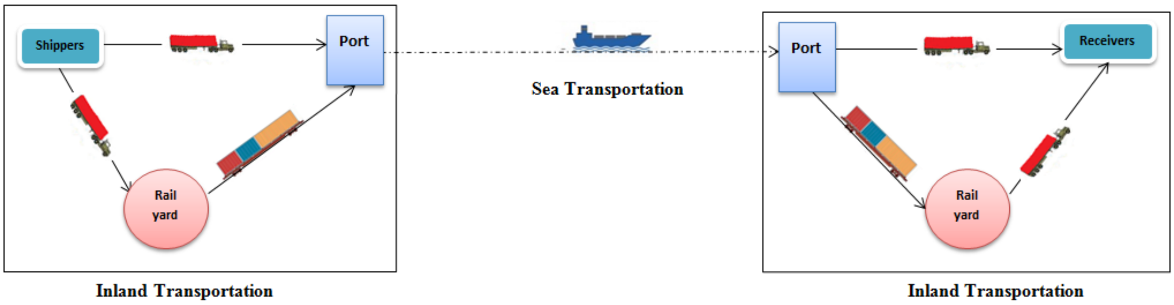

Figure 1.3: *The process of Intermodal Freight Transportation*

Sea (ocean) transportation is required for the case of global transportation among countries and far regions [14]. Many factors contribute to the importance of intermodal freight transportation [13] such as, the transportation cost from the economic perspective, the environment impact and traffic flows. For instance, as reported by [19], developed intermodal systems of transportation can decrease the emissions by 57% comparing to unimodal systems of transportation . As a result, intermodal freight transportation has attracted attention from academic researchers, business firms and governments to look at this operation carefully.

1.3 Inland (Drayage) Transportation of Containers

In contrast to sea transportation, inland transportation, which is also called drayage transportation as defined by Harrison et al. [49], is executed by rail and trucks. When the vessel arrives at the port, containers can be transported by rail, which is usually located at the port, to the final rail hub. From the rail hub, containers are then delivered by trucks to customers and vice versa. Another common type of inland transportation is that containers are delivered directly from the port to customers by trucks [11]. In this domain, shipping companies usually manage hundreds of the inland transportation orders every working day. The shipping firms aim to execute all of their orders with smallest cost whilst satisfying customer requirements. However, the inland transportation takes a significant portion of the total cost that arises from intermodal transportation. Therefore, it is important to create an efficient strategy to manage this process in a way to ensure all parties are satisfied. Below are examples of the inland transportation of orders which are planned by shipping companies and executed by haulage contractors by trucks and rail.

An import by truck The first type of order is called an import order, which is the delivery of loaded containers from the port/terminal to the destined customer location by trucks. After stripping the container at customers, the empty container must be taken to a suitable location for future use, such as a nearby depot or back to the original port (Figure 1.4). Note that if containers and trucks are not separated after the service, it is called a *Discharge* of containers. In contrast, if containers and trucks are separated after the service, this is called a *Strip* of containers. A practical UK example: A truck drives a loaded container from the port of Felixstowe to a customer in Cambridge, then the empty container is sent to a depot or directly to another customer or back to a container yard in the port of Felixstowe.

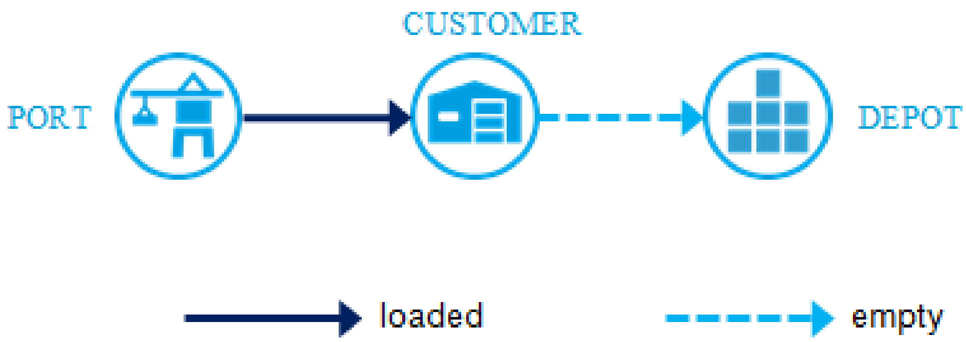

Figure 1.4: *An import of orders by truck*

An export by truck In opposite to the import of orders, export orders are executed by trucks following the same sequence in reverse, namely that a suitable empty container is picked from a depot or a port, driven to the customer location to collect the cargo, and then the loaded container is delivered to the port to be placed on a departing ship (Figure 1.5). A practical UK example: A truck picks up an empty container from a Northampton depot to collect a cargo at a customer in Bedford and takes the loaded container to the port of Felixstowe.

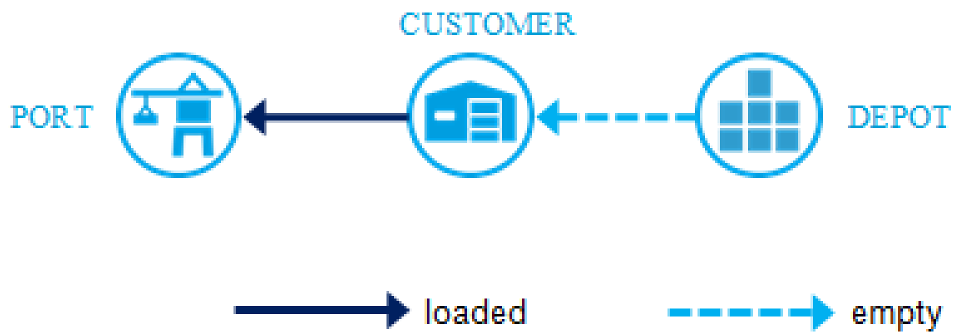

Figure 1.5: *An export of orders by truck*

An order by rail and truck It is often cost saving to transport loaded containers for most of the distance on a rail. For import orders delivering mostly by rail from the port will save a long distance left to the customer. From the rail hub, containers are then delivered to customers by trucks (Figure 1.6). The empty containers are then returned to the rail hub by trucks for reuse. Of course, the export order executed by rail and truck follows the reverse

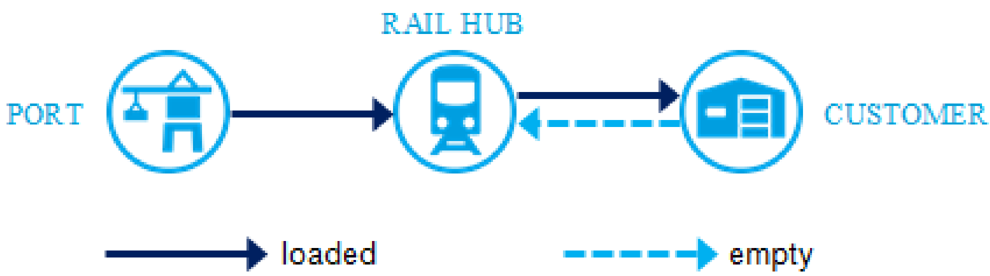

Figure 1.6: *The delivery of orders by rail and truck*

sequence. A practical UK example: A loaded container arriving on a ship at the port of Felixstowe is delivered by a rail service for Birmingham. Then a truck collects the loaded container from the Birmingham rail terminal to a customer in Derby and then returns the empty container to the Birmingham rail terminal.

A combination of import and export It is also cost saving to combine an import order with an export order in a matching case. In this scenario the empty container resulting from the import customer is then delivered directly to the export customer, finally the loaded container is delivered from the exporter customer back to the port (Figure 1.7). This type of transportation is called *Street-turn* of empty containers. In this case, two empty journeys from/to a depot/port are saved, in addition to the saving resulted from using a single truck to execute two trips of orders. In contrast, there is another common empty container strategy called *Depot-turn*, with which empty containers are delivered to depots for storing and reuse. A practical UK example: A truck collects a loaded container from a terminal at the port of Felixstowe to an import customer in Cambridge, then the empty container is sent to an export customer in Bedford, finally the loaded container is delivered back to the port of Felixstowe.

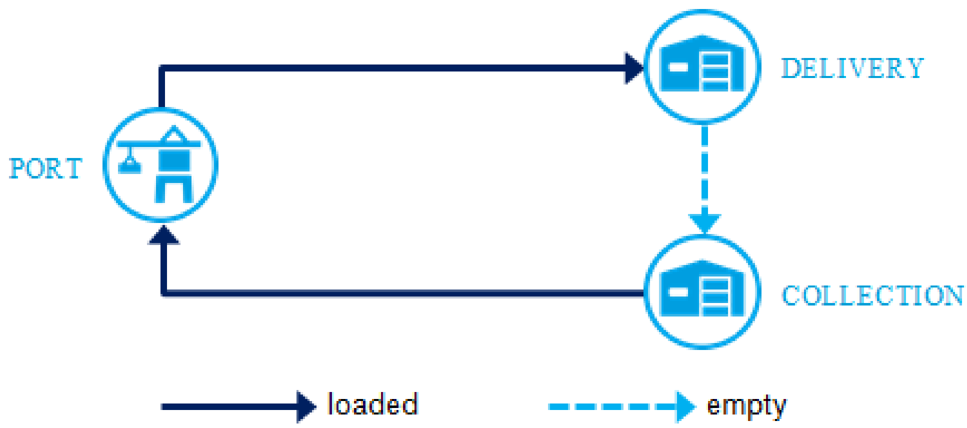

Figure 1.7: *A combination of import and export orders*

Pairing 2×20ft containers The pairing of 2×20ft containers means that two loaded/empty containers are paired in delivery by a single truck between the port and their destined customer locations. This problem is linked to both shipping lines and haulage companies as a carrier, as the aim of both parties is to save cost from the combination of orders. In the UK and Europe, two 20ft containers can be carried on a single 40ft truck [71]. The pairing process is typically performed on a slider chassis, which is able to carry either 2×20ft containers or 1×40ft container as shown in Figure 1.8. From the perspective of shipping companies, pairing two orders on a single truck can be cheaper than paying for two trucks to carry them out separately. While for the haulage companies combining orders from two separate shipping companies will earn the rate cost for both orders separately but only having the fuel and resource cost of a single truck. Note that, the pairing process is subject to certain constraints, such as the gross weight and driving time regulations. Combining orders in twinned pairs will naturally reduce the number of trucks required to execute the transport plan. However, there is a further measure of truck utilisation to consider. Each trip will have a duration and the shorter the trip, the more likely that the truck can do other work that day. Therefore it is worth considering how to minimise individual and overall durations alongside the main task of minimising cost.

Figure 1.8: *Pairing 2×20ft containers on 40ft truck (https://en.wikipedia.org/wiki/Long-combination-vehicle, 10/08/2018)*

1.4 Research Objectives, Scope and Organization

According to the United Nations Conference on Trade and Development [80] in 2014, around 80% of the global trade volume is transported by sea and handled by ports. Consequently, the demands of transporting containers from the port terminal to the receivers are high and increasing as the years pass [20]. The most significant cost related to the total transportation cost is caused by inland transportation, which illustrates the importance of the inland transportation of containers. Indeed, the unplanned use of trucks (lorries) will cause a high transportation cost, environment problems related to emissions increasing and also traffic flow problems. For these reasons, it is important to design efficient techniques and strategies to plan and manage this operation effectively. From the perspective of operational research and decision science, this thesis intends to develop efficient optimization models to tackle the container inland transportation problems. Many research efforts focused on solving these problems, however, several gaps in the current literature are noticed, which this thesis aims to cover to some extent. This thesis comprises six chapters as follows:

Chapter 1 is an introduction which provides an overview of the usage of containerization in logistics, a background to intermodal freight transportation, a description of the inland transportation of containers and finally the objectives and the scope of the thesis.

Chapter 2 investigates previous literature. Based on this investigation the literature review is separated into three categories. The exact and heuristic (metaheuristic) methods are also explained in this chapter.

In Chapter 3, we investigate the Pairing of Containers/Orders in Drayage Transportation (PCDT) of homogenous container types on heterogeneous fleets. More specifically, we consider the delivery of paired containers on 40ft trucks and/or individual containers on 20ft trucks, between a single port and a list of customer locations. An assignment Mixed Integer Linear Programming (MILP) model is formulated, which solves the problem of how to combine orders in delivery to save the total transportation cost when orders with both single and multiple destinations exist. In opposition to the traditional models relying on the Vehicle Routing Problem with Simultaneous Pickups and Deliveries and Time Windows (VRP-SPDTW) formulation, this model falls into the assignment problem category which is more efficient to solve on large size instances. Another merit for the proposed model is that it can be implemented on different variants of the container drayage problem: import only, import-inland and import-inland-export. Results show that in all cases the pairing of containers yields less cost compared to individual delivery and decreases empty tours.

In Chapter 4, we design a Mixed Integer Linear Programming (MILP) model for combining orders in the inland, haulage transportation of containers. In this MILP model, the pick up and delivery process of both 20 and 40 foot containers from the terminals to the customer locations and vice versa are optimized using a heterogeneous fleet consisting of both 20ft and 40ft trucks/chasses. Important operational constraints such as the time window at order receivers, the payload weight of containers and the regulation of the working hours are considered. Based on an assignment model, this MILP solves problems with 100 orders efficiently to optimality. To deal with larger instances, a decomposition and aggregation heuristic method is designed. The basic idea of this approach is to decompose order locations geographically into fan-shaped sub-areas based on the angle of the order location to the port, and solve the sub problems using the proposed MILP model. To balance the fleet size amongst all subgroups, column generation is used to iteratively adjust the number of allocated trucks according to the shadow-price of each truck type. Based on decomposed solutions, orders that are "fully" combined with others are removed and an aggregation phase follows to enable wider combination choices across subgroups. The decomposition and aggregation solution process is tested to be both efficient and cost-saving.

In Chapter 5, we reflect the real practice of haulage container transportation where both pick-up and delivery, empty and loaded, *Discharge* and *Strip* of heterogenous container types are combined. Heterogeneous fleets are considered to perform the inland transportation. For managing the delivery of empty containers, the two common strategies, *Depot-turn* and *Street-turn* are both tested with examples capturing real geographical information. A Genetic Algorithm (GA) approach is designed for solving large scale problems. The result shows that solving the complicated problem using the developed GA is better than solving the simplified problem using an exact method, even on small scale instances where an optimal solution is achievable with the exact model. In addition, combining the *Strip* and *Discharge* types with the usage of inland empty depots both saves transportation cost and increases fleet utilization. The value of using inland depots is also evaluated.

In Chapter 6, conclusions of the research are drawn and the future research recommendations are discussed.

To summarize, this thesis fills some gaps between the current academic study and the industrial operations of container inland transportation, and contributes to the development of heuristic approaches to solving large-scale problems.

Chapter 2

Literature Review

Many studies have been established to consider the inland transportation of containers and the majority of them focus on the usage of Operational Research/Optimization techniques. A wide range of descriptions and classifications were introduced by Steenken et al. [93] about the optimisation methods that had been used in the container terminal processes, which was later expanded and updated by Stahlbock and VoB [92]. On the other hand, Braekers et al. [5] described in more detail the operation of managing the transportation of empty containers at multiple planning levels (strategic, tactical and operational). In this chapter, we review the previous literature that addressed the container inland transportation problems, i.e. the haulage pickup and delivery of containers. As we can see in Figure 2.1, problems considered in this thesis are classified into three types, and accordingly the literature review is categorized into three main subsections: homogenous container sizes, heterogenous container sizes, strip and discharge of containers.

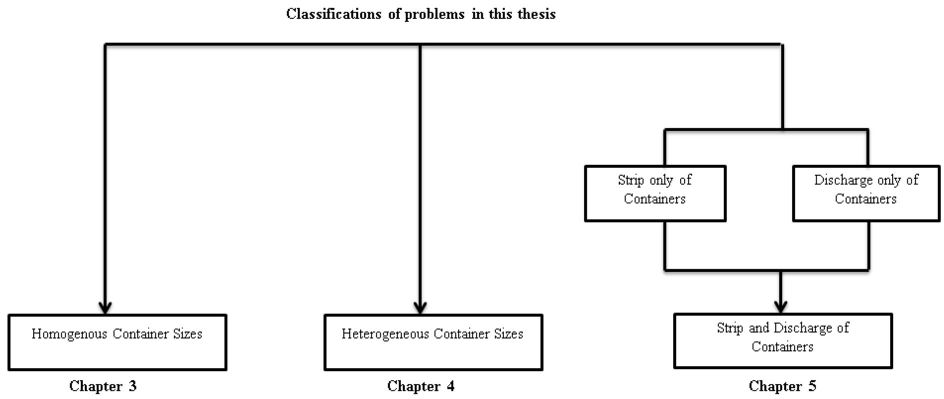

Figure 2.1: *Classifications of problems of this thesis*

2.1 Homogenous Container Sizes

Due to its importance, inland transportation of containers has received attention from both academia and industry over the last few years. In the container industry, the most common types of containers used are the 20ft and 40ft containers. Therefore, these two types are considered by most articles about inland transportation. In the existing literature, some studies optimized the movement of containers of the same size. In this case, the delivery and pickup of homogenous containers, 20ft or 40ft, were investigated. The aim was to combine the delivery and pickup services so as to save overall transportation cost. However, some articles focused only on loaded containers, while others focused on loaded and empty containers. Except one paper [7] who studied the pairing of two 20ft containers on one 40ft chassis, a large amount of literature focused on the transportation of a single container per-truck. Note that the problem description in some articles were the same and the difference was only seen in the developed methodology.

Jula et al. [56] introduced a transportation problem, where a set of loaded containers should be transported to meet a list of pickup and delivery requests. For the same overall transportation cost, pickup and delivery orders are paired before being assigned to delivery routes. By allocating a single node for each pickup and delivery pair, a multiple traveling salesman problem with time window constraints (M-TSPTW) model was formulated. This model solved limited size instances. If the problem became too large to handle with this method, a hybrid method was then used, formed of dynamic programming techniques in conjunction with genetic algorithms. A heuristic insertion method, which was updated from Jaw et al. [54], was also developed and solutions obtained from this method were compared with the M-TSPTW and hybrid method solutions. The results showed that the heuristic insertion method outperformed the M-TSPTW and hybrid method. Later, Jula et a. [55] studied the delivery of a single class of empty containers in the Los Angeles and Long Beach port. A mathematical model was formulated and solved by a two phase optimization method on different simulation scenarios. Imai et al. [53] studied the same problem as that of Jula et al. [56]. However, in this research it assumed that the haulage company have two types of truck, one type owned by the company and the other rented from outside. The aim was to make use of the own fleet as much as possible so as to reduce the cost paid for rental. The problem was formulated as a vehicle routing problem with full container load (VRPFC), in which a homogenous size and type of containers was used. According to the authors, the VRPFC was NP-hard, therefore a subgradient heuristic based on a Lagrangian relaxation was developed to identify the near optimal solution for large sizes instances. In the following years, Caris and Janssens [8, 9] investigated the same problem as in Imai et al. [53]. In order to solve the problem, the authors proposed a two-phase method, where the initial solution was constructed by an insertion heuristic. This initial solution was then improved with local search techniques based on three neighbourhoods and deterministic annealing approaches. Zhang et al. [107, 109] extended the work of

Imai et al. [53] and Caris and Janssens [8, 9], by introducing two types of empty orders (import and export empty containers) of the container truck transportation (CTT). The problem was formulated mathematically as an extension of the multi travelling salesman problem with time window (M-TSPTW) based on window partition [101]. A cluster and reactive tabu search (RTS) method was developed to solve the problem. Later, Zhang et al. [110] modified their previous research by restricting the number of empty containers stored at depots to a limited number. The problem was also formulated as an extension of the M-TSPTW based on the direct graph and solved by the reactive tabu search to create routes visited by trucks. Similar to Zhang et al. [110], Nossack and Pesch [74] formulated the same problem as a full truckload pickup and delivery problem with time window (FTPDPTW) and solved by a two stage heuristic method. In the first stage, a route construction heuristic was built. Then, another heuristic method was developed to improve the constructed route. Braekers et al. [3] studied a similar problem to that of Nossack and Pesch [74], however in this case empty containers were assumed to be delivered directly from suppliers to demanders without the existence of inland depots. The problem was formulated as an asymmetric multiple vehicle traveling salesman problem with time window (am-TSPTW) followed by a sequential and integrated heuristic method based on a deterministic annealing algorithm. For two different scenarios, Kopfer et al. [61] studied the delivery of homogenous containers. The first scenario is when empty containers are reused by the same owner, and the second is the sharing of the empty containers between more than one owner. Two mathematical models were formulated to solve these two scenarios. Later, Braekers et al. [4] considered and solved the same problem as a bi-objective problem (minimizing distance and number of trucks) based on two phases hybrid deterministic annealing and tabu search approach. Recently, Song et al. [91] adapted the Braekers et al. [4] approach by solving the problem as an asymmetric vehicle routing problem with time window (a-VRPTW) based on an arc-flow formulation and a branch-and-price-and-cut algorithm. Sterzik and Kopfer, [94] imposed a mixed integer programming (MIP) model for the inland container transportation. This model varies from others in which it considered vehicle routing, scheduling and allocating empty containers. According to the authors, this problem was an extension of the pickup and delivery problems with time window (PDPTW) in which trucks can be allocated to the nearest depot after completing the service when not all pickup and delivery nodes are known in advance. A tabu search heuristic was proposed to solve the problem. For different problem set-ups, Wang and Yun [100] addressed the transportation of containers by trucks and trains. A mixed integer programming model followed by a hybrid tabu search method was designed to solve the problem. Xue et al. [104, 105] examined the separation of trucks and trailers for the delivery and pick of containers. Mathematical models were formulated and solved by ant colony optimization and tabu search methods. Schulte et al. [86] introduced a collaborative planning model to be operated within a truck appointment system to investigate its impact on emission and cost objectives. An optimization model based on the M-TSPTW was developed.

On the other hand, the pairing of two 20ft loaded/empty containers on a 40ft chassis was investigated in the paper of Caballini et al. [7]. A mathematical model for combining the import, export and inland transportation trips was developed. The aim of combining trips is to reduce the empty trips and the total cost.

Having investigated these articles and building on their shortfalls, the aim of Chapter 3 [23] of this thesis is to further minimize the inland transportation cost of 20ft containers by allowing the simultaneous transportation of two 20ft containers on 40ft trucks following UK regulations. This significantly reduces the empty movements of the trucks on their routes whilst respecting time windows and laws which must be abided by. Here we propose a generic optimization model based on an assignment problem (rather than the traditional VRP ideas), which can be adapted to the combination of containers in three following ways: import (export) and inland; import and export; and import-inland-export.

2.2 Heterogenous Container Sizes

In addition to the transportation problem of homogenous container sizes, the heterogenous container types combination problem in road transportation has also been studied widely. This type of problem normally refers to the delivery of both 20ft and 40ft containers using homogenous or heterogenous fleets. In this case 40 ft trucks can carry up to two 20ft containers or one 40ft container. Some literature studied the problem from the perspective of managing the delivery of loaded and empty containers amongst their origins and destinations in order to minimize the transportation cost, while other literature considered fleet management aiming to minimize the number of required vehicles or to maximize fleet utilization.

Chung et al. [16], formulated a mathematical model for the pickup and delivery of both 20ft and 40ft containers as a single and multi-commodity problem. In a similar fashion, Vidovic et al. [98, 99] suggested to construct an optimal delivery plan by merging the pickup and delivery task nodes into executable routes. A matching mathematical model was formulated based on this idea which can be used for small size instances. For larger problems, a variable neighbourhood search heuristic was proposed. Zhang et al. [108] investigated the same problem using a multiple traveling salesman model consisting of three tree search and an improved reactive tabu search algorithm. In another article, Wen and Zhou [102] investigated the pickup and delivery of loaded and empty containers as a local container vehicle routing problem with variable travelling time. To solve the problem, an integer programming model followed by a genetic algorithm method were formulated. In his thesis, Lai [62] addressed the delivery problem of a single or two loaded/empty containers using homogenous and heterogeneous fleets of trucks. The problems were presented as a vehicle routing problem with backhauls (VRPB). Then, an integer linear programming (ILP) model was formulated and solved by an efficient metaheuristic approach. The metaheuristic method constructed the initial solutions based on the Clarke and Wright method, and improved it by several local search phases. Similarly, Lai et al. [64] proposed a metaheuristic method that reduced the travelling distance of routes by interchanging routes and nodes. The metaheuristic method also consisted of two phases, the first phase was based on the Clarke and Wright method to create the initial solution, while the second phase was based on a local search method to improve the initial solution. Later, Lai et al. [63], adapted their previous research by formulating the problem using an integer linear programming model and solved it using an adaptive guidance metaheuristic. Schönberger et al. [85] investigated the pick up and delivery of loaded and empty containers, however source/destination of empty containers in this case were assumed to be unknown in advance. A mixed integer linear programming model, which is an extension of the pickup and delivery problem (PDP) with less than truckload (LTL) orders was formulated to solve the problem. The model involved different types of stages comprising: deciding the pickup and delivery locations, routing of empty containers and routing of trucks. Popovic et al. [79] also considered the pickup and delivery of empty and loaded containers by demonstrating a variable neighbourhood search (VNS) heuristic to solve container drayage problems considering time windows (CDPTW). Similarly, Funke and Kopfer [34] studied the transportation of heterogenous containers by improving a neighbourhood search (NS) technique to optimize the container routing and scheduling. Later, Funke and Kopfer [36] duplicated nodes within the graphical representation to identify customers that are needed to be visited more than once. They then used a multi-commodity flow model with multiple travelling salesperson ideas to ensure that each container movement was covered and to minimize the distance and time travelled by trucks on their routes.

In contrast, Reinhardt et al. [82] studied the delivery of single import and export orders, in which each truck can only carry a single 20ft or 40ft container. Street turns strategy was applied in which empty containers were delivered directly from import customers to export ones to load and send it back to the port. An integer linear programming (ILP) model was formulated. In the following years, Reinhardt et al. [83] adapted their previous work by applying different mathematical models based on the vehicle routing and scheduling problem. Many scheduling problems, such as balancing empty containers for multiple storage depots and optimising the fleet size were investigated. For a different setup problem, Tan et al. [96] built a model for the truck and trailer vehicle routing problem (TTVRP) delivering loaded and empty containers. In this case trucks and trailers were assumed to be separated in different locations and the aim was to minimize the travelling distance and the number of used trucks. A hybrid multiobjective evolutionary algorithm in conjunction with genetic operators, variable length representation and local search heuristic were developed to find solutions for the TTVRP. Nordsieck et al. [73] investigated the drayage operations at a marine terminal by making use of the street turns strategy to deliver empty containers directly from importers to exporters. A heuristic method was developed to solve the problem. Recently, Ghezelsoflu et al. [38] addressed the heterogenous drayage problem with a set-covering model which only allows to visit up to four locations.

On the other hand, some literature studied inland transportation from the prospective of fleet management. Wang and Regan [101] developed a multiple traveling salesman problem with time window constraints (m-TSPTW) for local truckload pickup and delivery problems. An iterative method consisting of an over-constrained and an under-constrained scheme and a specific time window partitioning scheme was described. In a similar fashion, Gronalt et al. [44] studied the pickup and delivery of full truckloads between customers considering the time windows (PDPTW). A relaxed problem formulation to estimate the lower bound was presented and four different saving heuristics were proposed. Francis et al. [33] presented a multi-resource routing problem (MRRP) method to model the drayage operations in which the two resources, tractors and trailers, were assigned to perform the delivery of loaded and empty containers. A variable radius method (VR) and greedy randomized procedure were developed to solve the problem. These methods managed the number of options considered for flexible tasks (either feasible origins for a known destination or feasible destinations for a known origin). In contrast, both Coslovich et al. [18] and Smilowitz [90] focused on the management of a fleet of trucks considering the resource costs (drivers and trucks), the routing costs and the container repositioning cost. Coslovich et al. [18] formulated an integer programming problem and solved it based on the decomposition of the problem into three simpler subproblems associated to each type of the considered costs. Smilowitz [90] presented the problem as asymmetric vehicle routing problem, which was adapted from Bodin et al. [2], and solved it based on column generation embedded in a branch and bound method. Similarly, Cheung et al. [11] studied the managing of drayage activities in the Hong Kong port. The authors claimed that managing drayage operations was difficult since many resources such as drivers, tractors, and chasses are required to be managed simultaneously. An attribute decision model was formulated for this problem and solved by an adaptive labeling algorithm. Namboothiri and Erera [72] studied the problem of managing a fleet of trucks and provided a transportation service which included the pickup and delivery of containers to a port with an appointment access system. A planning system based on an integer programming model was developed. The model comprised several stages: the feasible customer requests were firstly determined, followed by the best schedule vehicle routes, and finally the vehicle numbers were minimized comparing to all available vehicles. To solve the problem, a heuristic method based on column generation was developed. Shiri and Huynh [88] extended the study of Namboothiri and Erera [72] by addressing the drayage assignment problem, which involved the delivery and pickup of import and export containers. A mathematical model which was an extension of the multiple traveling salesman problem with time windows (M-TSPTW) was formulated. In addition, an algorithm based on reactive tabu search (RTS) was created to solve the M-TSPTW model. Later, Shiri and Huynh [89] designed a drayage scheduling model which assessed the U.S. chassis supply models. In this model trailers, trucks and containers are separated in different locations, and the aim in this case was to ensure that containers and trailers were of the same sizes. A mixed integer quadratic programming model, which was an extension of the M-TSPTW, was formulated and solved by a reactive tabu search (RTS) algorithm combined with an insertion heuristic.

Falling into the category of the delivery of heterogenous container types, the aim of Chapter 4 [45] of this thesis is to develop a mixed integer linear programming (MILP) model for combining orders for the pickup and delivery of both 20ft and 40ft containers by a heterogeneous fleet. However, in this research the aforementioned literature is extended by considering more realistic restrictions and combination possibilities based on a more efficient assignment model structure. Unlike the PDP model structure that is commonly used to tackle container delivery issues, the assignment-based model solves much more efficiently, which finds optimal solutions for larger problems in several hours. For the purpose of solving large size instances and balancing fleet size, a decomposition and aggregation heuristic, based on the column generation approach, is designed.

2.3 Strip and Discharge of Containers

Historically, most container inland transportation literature considers one type of loading/unloading rule out of *Strip* and *Discharge*. We therefore organise the literature review accordingly. In the *Strip* case, a container (loaded or empty) is put onto/removed from the carrying truck at a customer location. Literature considering the Strip only of containers is summarised in Table 2.1. Most of it has already been explained in Subsections 2.1, 2.2.

Literature	**Type of Containers**	**Methodology**
Wang and Regan [101]	homogenous	Heuristic
Jula et al. [55,56]	homogenous	Heuristic
Chung et al. [16]	heterogenous	Exact
Francis et al. [33]	homogenous	Heuristic
Wen and Zhou [102]	heterogenous	Heuristic
Zhang et al. [107,109,110]	homogenous	Heuristic
Zhang et al. [108]	heterogenous	Heuristic
Kopfer et al. [61]	homogenous	Heuristic
Vidovic et al. [98,99]	heterogenous	Heuristic
Braekers et al. [3,4]	homogenous	Heuristic
Sterzik and Kopfer [94]	homogenous	Heuristic
Wang and Yun [100]	homogenous	Heuristic
Nossack and Pesch [74]	homogenous	Heuristic
Xue et al. [104,105]	homogenous	Heuristic
Song et al. [91]	homogenous	Exact

Table 2.1: *Literature considering Strip only of orders*

As we can see from Table 2.1, some literature formed their research around the Strip of loaded/empty homogenous containers, while others considered heterogenous container sizes. Accordingly, various mathematical models and heuristic/metaheuristic methods have been developed to tackle the Strip of containers. The literature which investigated the Strip case claimed that this approach is cost and time saving, since that instead of staying with containers during the service, trucks can leave to execute another trip and later empty containers can be removed by another truck/trip.

On the other hand, in the *Discharge* case, containers and trucks are not separated during/after the service. From this perspective, several articles that considered the Discharge only of containers are summarised in Table 2.2. Again most of them have been illustrated

Literature	Type of Containers	Methodology
Gronalt et al. [44]	homogenous	Heuristic
Tan et al. [96]	heterogenous	Heuristic
Imai et al. [53]	homogenous	Heuristic
Namboothiri and Erera [72]	homogenous	Heuristic
Caris and Janssens [8]	homogenous	Heuristic
Lai et al. [63, 64]	heterogenous	Heuristic
Reinhardt et al. [83]	homogenous	Exact
Shiri and Huynh [89]	homogenous	Heuristic
Schulte et al. [86]	homogenous	Exact
Ghezelsoflu et al. [38]	heterogenous	Exact

Table 2.2: *Literature considering Discharge only of orders*

in Subsections 2.1, 2.2. As shown in Table 2.2, some literature developed exact/heuristic methods for the homogenous containers and others studied the heterogenous type. These articles that considered the Discharge case argued that if a truck stays with a container during the service to collect it when emptied, this will save an additional trip/truck. In addition, this case is more convenient for customers, since drivers will take care of the cargo and the container and will follow customer instructions during the service.

As we can see from Table 2.3, only four articles have investigated both cases of *Strip* and *Discharge*. Ileri et al. [52] and Choi et al. [12] considered the problem of container transportation, where trucks (tractors) and chasses (trailers) are allocated to different depots. Ileri et al. [52] formulated a set partitioning model based on column generation for the delivery of homogenous containers, while Choi et al. [12] developed a genetic algorithm approach to solve a heterogenous container case. Chapter 5 of this thesis is different from Ileri et al. [52] as we investigated the heterogenous container types, and is different from Choi et al. [12] as

Literature	Strip	Discharge	Type of Containers	Methodology
Ileri et al. [52]	✓	✓	homogenous	Exact
Choi et al. [12]	✓	✓	heterogenous	Heuristic
Zhang et al. [106]	✓	✓	homogenous	Heuristic
Funke and Kopfer [35]	✓	✓	heterogenous	Exact

Table 2.3: *Classification of Literature based on Strip and Discharge of orders*

they considered only the inland depots for managing empty containers, while we studied both cases of *Street-turn* and *Depot-turn*. In the paper of Zhang et al. [106], the drayage problem of homogenous containers was studied. A mixed integer nonlinear programming model based on a determined-activities-on-vertex (DAOV) graph and a number of strategies including a window partition based (WPB) strategy were displayed. Later, Funke and Kopfer [35] adapted Zhang et al. [106] by formulating a mixed integer programming model for the delivery of heterogenous containers. Nevertheless, Funke and Kopfer [35] used homogenous trucks (only 40ft trucks) to perform the delivery whereas we use the two types of 20ft and 40ft chassis. In addition, Funke and Kopfer [35] considered only a single depot, while we used more than one inland depot for empty containers.

Moving from this scenario, the aim of Chapter 5 [46] of this thesis is to expand our previous work in Chapters 3 and 4, in which we consider the individual Discharge and Strip, by developing a Genetic Algorithm (GA) approach for solving the individual and joint *Strip* and *Discharge* cases. For managing the delivery of empty containers, the *Street-turn* and *Depot-turn* cases are applied. The value of using inland depots will be investigated.

2.4 Methodology of the research

In Operational Research (OR) and Optimization, Combinatorial Optimization Problems (COP) represent an attractive field for the interest of researchers. This class of problems originates from the fact that many real world applications can be classified as a COP. The planning and managing of different resources, various types of machines and sometimes people require a decision making process to consider only integer (or binary) decisions. The combinatorial optimization techniques can be used to achieve the optimal or the best (near) solutions for this sort of problem, which aim to minimize the overall cost, maximize profits and utilization of resources [51]. In this research we deal with a specific class of COP, which is the inland transportation of containers as explained in Chapter 1 .

As explained by [38], optimization problems fall into the categories of discrete, continuous, mixed (discrete and continuous), linear and nonlinear, based on the type of variables and constraints of the problem. For example, a Linear Programming (LP) model is comprised of a linear objective function and constraints with continuous variables as follows:

$$Max/Min \;\; c^T x \tag{2.1}$$

$$s.t. \quad ax \leq b \tag{2.2}$$

$$x \geq 0 \tag{2.3}$$

where x is a vector of continuous variables.

Similar to the LP, an Integer Linear Programming (ILP) model has linear constraints and integer variables, while Mixed Integer Linear Programming (MILP) models, which we will consider in later research is a type of ILP in which some variables are integer while others are continuous. The general formula of the MILP model is as follows [38]:

$$Max/Min \;\; c^T x + g^T y \tag{2.4}$$

$$s.t. \quad ax + hy \leq b \tag{2.5}$$

$$x \geq 0 \quad and \;\; integer, \;\; y \geq 0 \tag{2.6}$$

where x in this case is a vector of integer variables and y is a vector of continuous variables.

Common mathematical models are developed to solve typical combinatorial optimization problems, for instance TSP, vehicle routing, job scheduling, set covering, Knapsack, etc. Many solution techniques and algorithms are designed to solve these sorts of problems. In general, combinatorial optimization methods can be classified as exact methods which are designed to achieve the optimal solution, and approximated methods such as heuristics or meta-heuristics with which a best (near) optimal solution is expected [51].

As reported in [28] there are two common types of combinatorial optimization problems: the first type is referred as $\mathcal{P}$, which indicates the problems that can be solved in polynomial time. This type of problems are characterised as easy to solve. While the second type is the non-deterministic polynomial time problems, which is referred to as $\mathcal{NP}$ and this type involves the problems where the algorithm is not able to find an estimated solution and ensure the obtained solution is actually optimal in polynomial time. Sub types of the $\mathcal{NP}$ class contains $\mathcal{NP}-complete$ and $\mathcal{NP}-hard$. The problem is called $\mathcal{NP}-complete$ if solutions are sufficient to deal with any other $\mathcal{NP}$ problem in polynomial time. While problems are called as $\mathcal{NP}-hard$, if the related problem that can be classified as a decision problem is $\mathcal{NP}$ – *complete*. In general, combinatorial optimization problems are real challenges for researchers in many disciplines. For solving some regular and small problems, exact methods can be applied as we will explain in Section 1.5, while for complex and large size problems, exact methods are unsuitable, thus approximated methods should be developed to solve this type of problem as we will explain in Sections 1.6 and 1.7.

2.5 Exact approaches (algorithms)

Exact algorithms are designed to find the optimal solution for combinatorial optimization problems. These approaches are applied successfully to achieve the optimal solution for problems with a small number of variables and constraints, or when special structure appears in the constraint matrix . Some exact algorithms are illustrated as below:

Branch-and-Bound: An optimization method suggested by Land [65] to solve (mixed) integer linear programming (ILP) models. This method is based on implicit enumeration search of all possible candidate solutions and discarding non candidate solutions. The process of discarding the non candidate solutions depends on the estimated upper and lower bounds of the problem that required optimization. Thus, nodes with less or higher objective function than the current best solution are ignored. The first step of this technique begins by solving the problem using LP relaxation, then the feasible region for the LP relaxation is branched in an attempt to find optimal solution of the (M)ILP. The process terminates when the lower bound meets the upper bound and decision variables obtained from the LP relaxation solution meet the integrality requirements, otherwise, the branching process continues.

The cutting-plane algorithm: This algorithm was introduced by Ralph Gomory in the 1950s to obtain solutions for integer programming and convex optimization problems. The basic idea of this method is to specify and add a set of linear inequalities which are called *cuts* to the formulated model to achieve the best integer solution to the problem. The algorithm starts by solving the relaxed linear programming problem to find an initial continuous solution, which will then be cut off with an integral constraint. After the linear constraint is added, the modified linear programming relaxation will be solved again. The procedure is repeated until an integer solution to the problem is obtained.

Branch-and-Cut: Another technique for solving integer linear programming (ILP) problems is called Branch-and-Cut [76], which is a combination of the Branch-and-Bound and the cutting plane algorithms. The basic idea of this algorithm is to tighten the LP relaxation by using the branch and bound and the cutting planes algorithms. The method starts by solving the LP model without the integer values using the simplex algorithm. In case the optimal solution is a non-integer value for a variable that is supposed to be integer, the cutting plane algorithm is then used to add a new inequality constraint, which is satisfied by all feasible integer points but violated by the current fractional solution, such that resolving it will yield a less fractional different solution. Next, the branch and bound is started by splitting the current problem into multiple versions. Then, the obtained LP is solved again by using the simplex method and the process is repeated.

Column Generation (CG): This method was first sugested by Ford and Fulkerson [32] in the context of a multi-commodity network flow problem and was adapted by Dantzig and Wolfe [24] to solve linear programming problems (LPs) with a decomposable structure.

Later, Gilmore and Gomory [40] demonstrated its effectiveness in solving the cutting stock problem. Some of LPs and ILPs arising in combinatorial optimization problems are intractable to solve and the reason is the large number of variables involved in the problem. The column generation procedure is based on a decomposition technique, for solving a structured linear program (LP) with few rows but many columns (variables). The column generation procedure decomposes the LP into a master problem and a subproblem. The master problem contains a subset of the columns, each of which is an optimal solution for the subproblems that have been solved. The subproblem, which is a separation problem for the dual LP, is solved to identify whether the master problem should be enlarged with additional columns or not. The column generation procedure alternates between the master problem and the subproblem, until the former contains all the columns that are necessary for finding an optimal solution of the original LP [27].

Generally speaking, column generation is a way of beginning with a small, manageable part of a problem solving and analyzing that partial solution to discover the next part of the problem to add to the model, and then resolving the enlarged model. Column generation repeats that process until it achieves a satisfactory solution to the whole problem.

In order to illustrate CG, assume there is a LP:

$$Min \;\; z = c^T x \tag{2.7}$$

$$s.t. \quad ax \geq b \tag{2.8}$$

$$x \geq 0 \tag{2.9}$$

Suppose that X is the domain of x, in the case of large size problems, it is difficult to include all the variables in the cardinality X, alternatively a subset such as $J \subseteq X$ is suggested and the restricted master problem (RMP) can be formulated as follows [27]:

$$Min \;\; z = \sum_{j \in J} c_j \lambda_j \tag{2.10}$$

$$s.t. \quad \sum_{j \in J} a_j \lambda_j \geq b \tag{2.11}$$

$$\lambda_j \geq 0, \forall j \in J \tag{2.12}$$

In this case, each column in set (J) is associated with a variable λ_j, indicating the number of times a column is chosen in the solution of the RMP. Let λ and π be the primal and dual optimal solutions of the RMP, the subproblem is then [27]:

$$z^* := Min\{c_j - \pi^T a_j | j \in J\} \tag{2.13}$$

We assume that this problem is feasible, for otherwise the master problem would be empty as well. If the subproblem solution is non-negative, namely no reduced cost coefficient has negative value, the solution λ to the restricted master problem optimally solves the master problem as well. Otherwise, we enlarge the RMP by the column derived from the optimal solution to the subproblem, and repeat with re-optimizing the RMP.

2.6 Heuristic approaches

The exact methods are usually unable to solve large size combinatorial problems, instead heuristic (approximated) methods are preferred to obtain the best or near optimal solution. The quality of the solution depends on the robustness and the performance of the created heuristic approach. Indeed, many combinatorial optimization problems are $\mathcal{NP} - hard$, which means that it is difficult to use an exact method to obtain the optimal solution. Alternatively, heuristic algorithms need to be designed to find a best (good) solution for these problems. The early classical heuristic methods for vehicle routing problems were proposed in the period between 1960 and 1990 [87]. For instance, the Clarke and Wright savings algorithm [17], which is one of the most well-known classical heuristic techniques for solving hard optimization problems. By assuming that a number of available vehicles are used to deliver a load from a depot to a number of points (locations), the algorithm is started by linking the selected two nodes (locations) and estimating the cost saving of linking the paired nodes. Then the cost savings are ranked descendingly, and the route is constructed by linking the node pairs based on the cost saving until all required routes are obtained. Another common classical heuristic method is called the insertion heuristic, which comprises two types: sequential insertion method [69] by inserting nodes (locations) sequentially to construct routes one by one , and the parallel insertion type [15] by constructing several routes in parallel. Recently, more modifications have been made to the classical heuristic methods, to create new heuristic techniques based on the type and complexity of the problem.

Clustering (Decomposition) Heuristic method As mentioned above, exact optimization approaches can only solve a limited size of problem with large computational times, thus it is necessary to develop a heuristic method to tackle these issues. Clustering (decomposition), is one of the common methods which is created to solve complex and large size problems. Several clustering techniques have been proposed in transportation, for instance, the sweep algorithm, which is also called fan-shaped clustering, was introduced first by Gillett and Miller [39]. The basic idea of this algorithm is to cluster customers geographically to reduce the size of the problem. The sweep algorithm starts by extending a straight line from the depot and the line is rotated clockwise until some customers are included in the first route based on the time and capacity of the vehicle. The sweep is continued to begin a new route with the last customer that was removed from the first route. The process is repeated until all customers are assigned to a specific route. In this case all constructed routes need to be started and ended at the same depot [103]. Another type of clustering which was suggested by Fisher and Jaikumar [31], is called seed-based decomposition. This method is started by choosing a customer location as a *seed* point [6] and linking each two customers in sequence to create a route. The process is repeated until all un-assigned customers are allocated to a specific route. In the rectangle-shaped clustering [21, 22], locations are partitioned into rectangular sectors. Based on the gravity of the sector, the total distance of travelling from the depot to that centre of gravity and then to customers can be estimated .

2.7 Meta-heuristic approaches

The term Meta-heuristic was originally created by Glover [42], which comprises a collection of methods for solving combinatorial optimization problems. This collection of methods are classified as high performance heuristics, which usually require less effort than designing a particular heuristic. Indeed, if they are implemented perfectly a near (good) optimal solution can be obtained in an acceptable solution time. However, the process of adapting the meta-heuristics to solve a particular class of problems is challenging. According to Gendreau [37] ″metaheuristics are divided into two categories: single-solution metaheuristics where a single solution (and search trajectory) is considered at a time, such as simulated annealing and tabu search methods, and population metaheuristics where a multiplicity of solutions evolve concurrently such as the genetic algorithm ″. In addition, Gendreau [37] emphasized that it is possible to differentiate within each type between the constructive meta-heuristics, in which the solution is assembled randomly and the improvement meta-heuristics, where solutions are modified iteratively.

Simulated Annealing (SA) This method was first suggested by Khachaturyan et al. [57, 58] and was improved and called Simulated Annealing (SA) by Kirkpatrick et al. [59] to converge to the optimal solution of combinatorial optimization problems. The name of simulated annealing was inspired by the process of annealing metallurgy, which is a method of heating and slow cooling of solids to reduce its disorder and energy and increasing its atoms size. This concept of slow cooling in a SA algorithm can be explained as a gradual decreasing in the probability of choosing inferior solutions as an explored solution. In metaheuristics the property of accepting the inferior solutions will reinforce the probability of reaching the global optimal solution extensively. In general, the basic idea of SA is as follows: randomly or using an appropriate method the algorithm adopts an initial solution and finds a neighborhood solution, which is close to the current solution and tests the quality of the new solution. Based on a specific probability, the decision will be made to replace the new solution with the old one, if it is the best; otherwise it will be removed in case it is worse than the current solution. For combinatorial optimization problems, simulated annealing requires a careful choosing of parameters and conditions such as the number of iterations and the termination criteria.

Tabu Search (TS) [41] suggested the initial idea of this method which he later called Tabu Search (TS) in [42]. TS is a metaheuristic search technique applied for solving various optimization problems. The term tabu originated from "*taboo*" which means forbidden. TS is based on local search to explore widely the neighborhood space to find the best solution. This method starts by creating an initial solution for the problem. Then, a neighborhood search is started to find a solution better than the initial solution. A *tabu list* is created to record the search moves and to avoid repetition of moves during the search process. The tabu list allows to remove part of the old moves and record a new search moves. The period of declaring moves of the tabu list is called *tabu tenure*. To avoid the preventing of exploring more areas of search, TS usually uses an *aspiration criterion*. More promising moves of search can be explored by using the aspiration criteria. It is worthwhile to use two important mechanisms called *intensification* and *diversification*. In the intensification process, the search for better solutions is intensified to encourage solutions close to the recent solution. While in the diversification a new search space of areas can be explored [26].

As referred in [43], a type of TS is called as *probabilistic tabu search*. In this type, a probability is assigned to the neighborhood search. As a reason, a high probability is provided to some important moves to reduce the cost of the solution, while, a low probability is provided for moves that are caused to repeat old states of moves. Another type of tabu search named as *reactive tabu search* was developed by Battiti and Tecchiolli [1]. An important feature of reactive TS is the ability of adapting the tabu list automatically through the search process based on the number of occurrences of old moves.

Variable Neighbourhood Search (VNS) Variable neighborhood search (VNS) was proposed by Hansen and Mladenović [47, 48]. The basic idea of this technique is to increase the neighborhood search of the search space. The process continues until a best solution is found or a stopping criterion is achieved. In the VNS, the size of the neighborhood is expanding up to a specific value. A new solution is produced using a process called *shaking* by randomly generating a point from the current neighborhood. Choosing the appropriate shaking process helps to change the solution without changing the important features of the solution, which will lead to find a new optimal solution gradually. For instance, in the travelling salesman problem, the shaking process is executed by swapping and exchanging the visited cities iteratively. In order to reach a local optimum, the search is intensified in the VNS.

Genetic Algorithm (GA) Genetic Algorithm (GA) is considered a meta-heuristic technique that arises from the evolution and natural selection process in cyclelife of creatures in nature. The initial idea of GA was first developed by Holland [50]. Later, the genetic algorithm became one of the popular optimization methods, which can be adopted to solve

$\mathcal{NP}-hard$ and difficult optimization problems. The GA procedure starts by creating an initial population of individuals called *chromosomes*, which represent the possible initial solutions of the problem. Each chromosome consists of a series pieces called *genes*, which usually represent parameters of the investigated problem. In order to evaluate the survival of individuals, a *fitness* value is developed, i.e., the objective function in optimization problems. Individuals with small/high (based on the objective of the problem) fitness values are then selected for the next reproduction process. The reproduction in GA starts by choosing two chromosomes called as *parents*, then an operator called a *crossover* is applied to create new offspring (*children*) by exchanging gene elements between the two parents. Another operator in GA is called *mutation*. In the mutation operator, genes of individual children obtained from the crossover are swapped randomly. In this case, a new population will be generated from the mutation process which will be combined with the old population to generate a new generation. The complete process of the GA is repeated until stopping criteria are achieved. The outline of the genetic algorithm is as follows:

- ***Initialization:*** Create random population of chromosomes (solutions).
- ***Fitness values:*** Evaluate the fitness values of chromosomes.
- ***Selection:*** A portion of the initial population solutions are selected for the next reproduction.
- ***Crossover:*** Crossover operator is applied for the selected parents to create new offspring (children).
- ***Mutation:*** Mutation operator is applied to the genes of the new children.

- ***Reproduction:*** Reproduce a new population by combining the resulting children obtained from mutation with the constructed initial population.

- ***Stopping Criteria:*** Repeat the algorithm until certain criteria are achieved.

2.8 Summary

In this chapter, we have reviewed the literature concerned with inland transportation for the three types of problems: Homogenous Container Sizes, Heterogenous Container Sizes and Strip and Discharge of Containers Problem. The exact methods such as LP, ILP and MILP, in addition to some algorithms (Branch-and-Bound, the cutting-plane algorithm, Branch-and-Cut and Column Generation) that have been used for solving the exact models are explained in this chapter. Heuristic approaches such as the clustering/decomposition method as well as metaheuristics approaches such as SA, TS, VNS and GA are also reviewed.

Chapter 3

Homogenous Container Sizes

3.1 Introduction

In accordance with the International Standard Organisation (ISO) specification, containers are classified into standard sizes in which 20- and 40-ft ones are the most commonly used. Most truck chassis are designed to carry them [71]. Cheung et al. [11] referred to the inland container transportation as the operation of moving loaded and empty containers amongst terminals, rail hubs, customers and depots, which is also called drayage as defined by Harrison et al. [49]. Under the influence of global integration, container transportation has grown impressively around the world during the last decades. Considering the large volume of inland container transportation demand and how costly it is, good management strategies which are efficient enough to be suitable for large industrial implementations are needed [70]. Although inland delivery covers a very short distance in the entire container transportation, it is not as economical. As reported by [60, 68] about 25 - 40% of the total transportation cost is accumulated in drayage, which is then raised by [75] to as high as 40 - 80%. This significant cost illustrates the importance of optimizing the inland delivery routes, especially with a focus on reducing the unproductive routes to relocate empty containers. Indeed, inefficient usage of trucks not only yields higher delivery cost and emissions, but also brings pressure on the operations of the port and introduces unnecessary traffic.

In order to reduce unnecessary traffic flow, most works in the field attempt to combine pickup and delivery trips together to reduce empty movements of containers [3]. These studies are then extended to the cases that further merge the route with inland deliveries [30], and/or consider the usage of dual-carriage trucks [95]. No matter what specific context is considered, almost all previous studies base their discussions around the general Mixed Integer Linear Programming (MILP) model for the Vehicle Routing Problem with Simultaneous Pickups and Deliveries and Time Windows (VRP-SPDTW), which is originally designed in generic vehicle routing literature [74]. The optimal decision tells which link should be travelled by which truck. This makes perfect sense in the case where a number of individual trips are combinable to form a single delivery/pickup route, but not as necessary for container delivery since the latter normally just allows the combination of no more than two (import only) or four (import and export) trips in one return route due to the capacity of the vehicle (dual-carriage).

On the other hand, as in the VRP-SPDTW model, one has to start from transforming the demand graph into one with a distinct node for every single task, the number of nodes and links are largely increased which increases the difficulty of solving the problem, and therefore makes the solution only solvable via heuristics. In comparison, Vidovic et al. [97] proposed an alternative way which formulates the trip combination problem as a multiple assignment model. This formula tries to merge customer requests (import and export) together to form full delivery routes and the optimal decision directly shows which container should be paired with which other for transportation. Since the number of containers to combine is no more than two if the truck can carry only one container at a time or four if dual-carriage, the decision variable is at most 4 dimensional in its index. Based on the observation of the authors, the multiple assignment model can be solved efficiently by commercial software for instances having 63 containers which is much larger than 19 with the VRP-SPDTW model. Our study furthers this idea by including more realistic restrictions on general practice, such as the working restriction for drivers, the ready time of containers at and/or the expected departure from the port, and more importantly, containers with multiple customer locations as its receivers. Although the last case is infrequent in practice, the inclusion of it makes the model more adaptable. Later in this chapter we will show how to make use of the multi-destination container term to extend the initial, import only model to solve import-inland and/or inland-export problems.

In this chapter, we firstly propose an optimization model for the Pairing of Containers in Drayage Transportation (PCDT). This model considers the joint delivery of import containers only, namely the container movements from the port to inland customers. This study is important in its own right as there are many countries, such as the United Kingdom, doing many more imports than exports so that the demands are not always balanceable to form round trips. In accordance with realistic situations, in the model we cover all major restrictions for the drayage service such as the empty leg transportation, the heterogeneous fleet size, the arrival time of vessels (containers), the time window restriction at customer locations and/or the port, the working time regulations etc. The aim of the model is to minimize total distance travelled by all vehicles used and the penalty paid for potential over time works by the truck driver. Major contributions of this initial model are twofold: first it allows one container to have more than one receiver; second the model is more efficient to solve than the traditional models based on pickup and delivery in vehicle routing networks and therefore allow more accurate solution for large problems with more than 300 containers. Considering the usage of multi-destination containers, we then extend our parameter definitions to make the aforementioned optimization model also applicable to the combination of import (export) with inland trips and also to import-inland-export problems without using dual-carriage trucks. Taking use of our model the solution difficulties for these three types of problems are similar, as there are no major modifications to the model itself but just to the interpretation of the input data.

The rest of the Chapter is structured as follows. In Section 3.2 the problem statement and the optimization model is described. A practical variant is demonstrated in Section 3.3, and numerical experiments are presented in Section 3.4. Section 3.5 is for the summary of findings.

3.2 Problem Statement and Optimization Model

We start the description of the problem by defining some terms that will be used later. The term *order* in this research is referred to as a customer request of delivering the content of a loaded container from its origin to destination (examples for import, inland and export orders are given in Table 3.5). Only 20ft containers are used for transportation, since the delivery of 40ft container can only be carried out by 40ft trucks therefore we can simply assume all of them go without pairing to reduce the problem size. Note that in this study, when we are talking about the delivery (pick up) of container, we mean Discharge, i.e. the delivery (pick up) of the cargo inside the container rather than the container itself. After discharging the container at customer locations (e.g. for import case), it should be transported to a final empty storage which is normally the port or an inland depot, unless it has been assigned to a specific final destination. Similarly, an empty container has to be collected from an empty storage in order to start an inland or export trip, if the trip is not performed right after an import delivery (so that we have an empty container to use on the truck). One order comes with an origin (where the cargo departs), one (single-) or two (multi-) destinations (where the receiver locates), the time window constraints at all relevant locations (when the branches open), the available time of the container (when it is ready to be collected), the payload weight (weight of cargo) and probably an assigned final destination (where the empty goes to).

For multi-destination orders, we follow the full-twin assumption that is introduced by [108], i.e., a truck that has begun to handle the first customer's location of the container, has to handle the second customer's location before the truck starts to carry out a new order. We assume all information are determined beforehand, including the visiting sequences of customer locations for multi-destination orders. Note further that the *order* is defined on a single container basis, namely if there is a customer request consisting of multiple containers, we have to split it into multiple orders with the same data and allow the customer to be visited by more than one truck. The model is developed from the perspective of haulage companies who own a certain heterogeneous fleet of trucks and chassis, collect orders from shipping lines and other customers and commit to make the delivery of containers in time to their destinations. Decisions are made on whether containers should be assigned to an individual trip that is executed by a truck that is able to transport one 20ft container, or whether two containers should be paired in a specific order and served by a truck that is able to transport two 20ft containers, as well as where to place the empty container after delivery. The aim is to minimize the total travelling costs of the whole fleet and the penalty costs for potential overtimes of trips. Note that we consider the working time regulation for drivers and penalty cost only on a daily basis. We assume there are adequate numbers of drivers employed so that no one will work overtime for more than once during a week.

3.2.1 Parameters and Definitions

In following sections we will formulate an assignment model whose results will inform how to serve every order. We will be starting from considering and defining parameters for the import orders only (loaded containers starting from the port), and then extend the data definition to cover the import-inland and the import-inland-export cases. In more detail, we assume that the fleet consists of H_1 20ft trucks and H_2 40ft trucks. For the sake of simplicity and clarity in modelling, we assume that a 40ft truck is not allowed to perform single container transportation although in practice it might be possible. We assume all containers/orders considered, denoted by $\mathcal{N}$, are allowed to be paired with another in transportation. We denote by $\mathcal{P}_1$ and $\mathcal{P}_2$ the sets of orders having single destination and multiple (dual) destinations, respectively. Containers in subset D_p are the ones with final empty destinations determined, whereas D_0 are the ones without. An import order (loaded container) $i \in \mathcal{N}$ is to be picked up from the port (L_0) after its available time (A_i), delivered to its receivers (L_i if single destination, L_i^1 and L_i^2 in the determined sequence if multiple destinations) individually or paired within predetermined time window ($[T_i^s, T_i^e]$ if single destination, $[T_i^{s1}, T_i^{e1}]$ and $[T_i^{s2}, T_i^{e2}]$ if multiple destinations), and dropped off at the agreed final empty destination D_i if there is one or at a nearby empty storage that is chosen from $\mathcal{M}_0$. The handling time at the port (h_i) refers to the time of loading the container on the truck, whereas the turnaround time at receivers (O_i) is the time taken to discharge the cargo from container i. We denote by W_i as the container i's payload weight (weight of cargo) and by V the weight of truck, chassis and the empty container. The gross weight limit, V_1^m for 20ft truck and V_2^m for 40ft truck, is applied to all delivery routes. In the case of haulage companies to do the work, costs are normally determined based on banded rates, increasing in a roughly linear fashion relative to distance. In this study, for a known list of locations to visit, the cost is captured by a linear function to the total travel distance including both the loaded legs and empty leg, which is denoted by ($f(., ..., .)$). In addition to the mileage cost, we also consider the potential penalty that the haulage company may have to pay for overtime workings. According to the EU regulation, if a driver works more than $T = 9$ hours then a penalty of C should be paid for any extra time working. However working longer than $T_{max} = 11$ hours in a day is strictly inadmissible.

Parameters

H_1: total number of lorries available for single container delivery.

H_2: total number of lorries available for paired containers delivery.

$\mathcal{M}_0$: set of port/depots/exporters as empty leg destinations.

$\mathcal{N} = \mathcal{P}_1 \cup \mathcal{P}_2 = \mathcal{D}_0 \cup \mathcal{D}_p$: set of containers, in which: $\mathcal{P}_1$ is the set of containers with single destination, $\mathcal{P}_2$ is the set of containers with multi destinations, $\mathcal{D}_0$ is the set of containers for which an empty leg destination is not yet determined and $\mathcal{D}_p$ is the set of containers for which the empty leg destination is a known port or depot or exporter.

$A_i, i \in \mathcal{N}$: available time of container (order) i for departure from the port.

$h_i, i \in \mathcal{N}$: handling time of container i at the port.

$O_i, i \in \mathcal{N}$: turnaround time at order i's customer location.

T: regular working hours.

T_{max}: the maximum possible working time for one shift that is allocated by the regulation.

C: penalty cost for extra working hours.

L_0: the port.

$L_i, i \in \mathcal{P}_1$: customer location for single destination container i.

$L_i^1, L_i^2, i \in \mathcal{P}_2$: two consecutive customer locations for multi destination container i.

$[T_i^s, T_i^e], i \in \mathcal{P}_1$: the time window during which the container i (single destination) is meant to arrive.

$[T_i^{s1}, T_i^{e1}], i \in \mathcal{P}_2$: time window in which the container i is meant to arrive at the first location.

$[T_i^{s2}, T_i^{e2}], i \in \mathcal{P}_2$: time window in which the container i is meant to arrive at the second location.

$D_i, i \in \mathcal{D}_p$: the empty leg destination that has been determined for orders in $\mathcal{D}_p$.

$W_i, i \in \mathcal{N}$: the weight of container i, which contains both the cargo and the container weight.

V: weight of the lorry and chassis and the empty container.

V_1^m: weight limit for individual delivery.

V_2^m: weight limit for paired delivery.

$f(., ..., .)$: travelling cost for a sequence of locations. We rescale the straightline distance by a constant factor to approximate the road distance between each pair of locations.

$t(.,...,.)$: travel time for a sequence of locations.

M: large number.

3.2.2 Decision Variables

In order to capture the entire features of the model, two types of decision variables are introduced as below. Binary variables x_{ijd} and y_{id} are the decision on how the container should be transported, paired or individually and following what sequence; binary variables z_{ijd} and u_{id} are introduced to identify if penalty cost should be paid for extra working time; Continuous variables $s_i, v_i, v_i^1, v_i^2, T_{id}, T_{ijd}$ are used to calculate the departure time of containers from the port, the arrival times at customer locations and the total working time of the paired and individual trips of the working plan.

Binary Variables

- $x_{ijd} = \begin{cases} 1, & \text{if containers } i \text{ and } j \text{ are delivered paired to their destinations (single or multi) on the same lorry and end at empty leg destination } d. \\ 0, & \text{otherwise} \end{cases}$
  $\forall i \neq j \in \mathcal{N}, \forall d \in \mathcal{M}_0.$
- $y_{id} = \begin{cases} 1, & \text{if container } i \text{ is delivered individually to its destination (single or multi) and end at empty leg destination } d. \\ 0, & \text{otherwise} \end{cases}$
  $\forall i \in \mathcal{N}, \forall d \in \mathcal{M}_0.$
- $z_{ijd} = \begin{cases} 1, & \text{if working hours for paired trip is higher than the regular working time} \\ 0, & \text{otherwise} \end{cases}$

  $\forall i \neq j \in \mathcal{N}, \forall d \in \mathcal{M}_0.$

- $u_{id} = \begin{cases} 1, & \text{if working hours for individual trip is higher than the regular working time} \\ 0, & \text{otherwise} \end{cases}$
  $\forall i \in \mathcal{N}, \forall d \in \mathcal{M}_0$.

Continuous Variables

- $s_i, \forall i \in \mathcal{N}$: departure time of container i from the port.

- $v_i, \forall i \in \mathcal{P}_1$: arrival time of one-destination container i.

- $v_i^1, v_i^2, \forall i \in \mathcal{P}_2$: arrival times of multi-destination container i at its two customer locations.

- $T_{id}, \forall i \in \mathcal{N}, \forall d \in \mathcal{M}_0$: total working time of the individual trip of servicing container i.

- $T_{ijd}, \forall i \neq j \in \mathcal{N}, \forall d \in \mathcal{M}_0$: total working time of the paired trip of servicing containers i and j.

3.2.3 The mathematical model

An assignment Mixed-Integer Linear Programming (MILP) model for the paired/individual delivery on 40ft/20ft chassis of import orders can be described as below. Note that this model is constructed for import orders only. In the next section we will discuss how this model can be implementable as well on import-inland and/or import-inland-export cases.

$$\min \sum_{i\in\mathcal{P}_1}\sum_{j\in\mathcal{P}_1}\sum_{d\in\mathcal{M}_0} x_{ijd}[f(L_0,L_i,L_j,d)] + \sum_{i\in\mathcal{P}_2}\sum_{j\in\mathcal{P}_1}\sum_{d\in\mathcal{M}_0} x_{ijd}[f(L_0,L_i^1,L_i^2,L_j,d)]+$$
$$\sum_{i\in\mathcal{P}_1}\sum_{j\in\mathcal{P}_2}\sum_{d\in\mathcal{M}_0} x_{ijd}[f(L_0,L_i,L_j^1,L_j^2,d)] + \sum_{i\in\mathcal{P}_2}\sum_{j\in\mathcal{P}_2}\sum_{d\in\mathcal{M}_0} x_{ijd}[f(L_0,L_i^1,L_i^2,L_j^1,L_j^2,d)]+$$
$$\sum_{i\in\mathcal{P}_1}\sum_{d\in\mathcal{M}_0} y_{id}[f(L_0,L_i,d)] + \sum_{i\in\mathcal{P}_2}\sum_{d\in\mathcal{M}_0} y_{id}[f(L_0,L_i^1,L_i^2,d)]+$$
$$\sum_{i\in\mathcal{N}}\sum_{j\in\mathcal{N}}\sum_{d\in\mathcal{M}_0} Cz_{ijd} + \sum_{i\in\mathcal{N}}\sum_{d\in\mathcal{M}_0} Cu_{id} \tag{3.1}$$

$s.t.$

$$\sum_{j\in\mathcal{N}}\sum_{d\in\mathcal{M}_0}(x_{ijd}+x_{jid}) + \sum_{d\in\mathcal{M}_0} y_{id} = 1, \forall i\in\mathcal{N}; \tag{3.2}$$

$$s_i \geq A_i + h_i, \forall i\in\mathcal{N}; \tag{3.3}$$

$$s_i \geq (A_j + h_j)\sum_{d\in\mathcal{M}_0}(x_{ijd}+x_{jid}), \forall i\neq j\in\mathcal{N}; \tag{3.4}$$

$$s_i - s_j \leq M*(1-\sum_{d\in\mathcal{M}_0}(x_{ijd}+x_{jid})), \forall i\neq j\in\mathcal{N}; \tag{3.5}$$

$$V + W_i\sum_{d\in\mathcal{M}_0} y_{id} \leq V_1^m, \forall i\in\mathcal{N}; \tag{3.6}$$

$$V + (W_i+W_j)\sum_{d\in\mathcal{M}_0}(x_{ijd}+x_{jid}) \leq V_2^m, \forall i\neq j\in\mathcal{N}; \tag{3.7}$$

$$v_i = s_i + \sum_{j\in\mathcal{P}_1}\sum_{d} x_{jid}[t(L_0,L_j,L_i)+O_j] + \sum_{j\in\mathcal{P}_2}\sum_{d} x_{jid}[t(L_0,L_j^1,L_j^2,L_i)+2O_j] +$$
$$(\sum_{j\in\mathcal{N}}\sum_{d} x_{ijd} + \sum_{d} y_{id})t(L_0,L_i), \forall i\in\mathcal{P}_1; \tag{3.8}$$

$$v_i^1 = s_i + \sum_{j\in\mathcal{P}_1}\sum_d x_{jid}[t(L_0, L_j, L_i^1) + O_j] + \sum_{j\in\mathcal{P}_2}\sum_d x_{jid}[t(L_0, L_j^1, L_j^2, L_i^1) + 2O_j] + (\sum_{j\in\mathcal{N}}\sum_d x_{ijd} + \sum_d y_{id})t(L_0, L_i^1), \forall i \in \mathcal{P}_2; \tag{3.9}$$

$$v_i^2 = s_i + \sum_{j\in\mathcal{P}_1}\sum_d x_{jid}[t(L_0, L_j, L_i^1, L_i^2) + 2O_j] + \sum_{j\in\mathcal{P}_2}\sum_d x_{jid}[t(L_0, L_j^1, L_j^2, L_i^1, L_i^2) + 3O_j] + (\sum_{j\in\mathcal{N}}\sum_d x_{ijd} + \sum_d y_{id})[t(L_0, L_i^1, L_i^2) + O_i], \forall i \in \mathcal{P}_2; \tag{3.10}$$

$$T_i^s \le v_i \le T_i^e, \forall i \in \mathcal{P}_1; \tag{3.11}$$

$$T_i^{s1} \le v_i^1 \le T_i^{e1}, \forall i \in \mathcal{P}_2; \tag{3.12}$$

$$T_i^{s2} \le v_i^2 \le T_i^{e2}, \forall i \in \mathcal{P}_2; \tag{3.13}$$

$$T_{id} \ge v_i + [O_i + t(L_i, d)] - s_i - M(1 - \sum_d y_{id}), \forall i \in \mathcal{P}_1; \tag{3.14}$$

$$T_{id} \ge v_i^2 + [O_i + t(L_i^2, d)] - s_i - M(1 - \sum_d y_{id}), \forall i \in \mathcal{P}_2; \tag{3.15}$$

$$T_{ijd} \ge v_j + [O_j + t(L_j, d)] - s_j - M(1 - \sum_d x_{ijd}), \forall i \in \mathcal{N}, j \in \mathcal{P}_1; \tag{3.16}$$

$$T_{ijd} \ge v_j^2 + [O_j + t(L_j^2, d)] - s_j - M(1 - \sum_d x_{ijd}), \forall i \in \mathcal{N}, j \in \mathcal{P}_2; \tag{3.17}$$

$$T_{id} \le T + M(u_{id}), \forall i \in \mathcal{N}; \tag{3.18}$$

$$T_{ijd} \le T + M(z_{ijd}), \forall i, j \in \mathcal{N}; \tag{3.19}$$

$$T_{id} \le T_{max}, \forall i \in \mathcal{N}; \tag{3.20}$$

$$T_{ijd} \le T_{max}, \forall i, j \in \mathcal{N}; \tag{3.21}$$

$$y_{iD_i} + \sum_{j \in \mathcal{N}} (x_{ijD_i} + x_{jiD_i}) = 1, \forall i \in \mathcal{D}_p; \tag{3.22}$$

$$\sum_{i \in \mathcal{N}} \sum_{d} y_{id} \leq H_1; \tag{3.23}$$

$$\sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{N}} \sum_{d} x_{ijd} \leq H_2; \tag{3.24}$$

$$s_i, v_i, v_i^1, v_i^2, T_{id}, T_{ijd} \geq 0, \forall i, j \in \mathcal{N}, \forall d \in \mathcal{M}_0; \tag{3.25}$$

$$x_{ijd}, y_{id}, u_{id}, z_{ijd} \in \{0, 1\}, \forall i, j \in \mathcal{N}, \forall d \in \mathcal{M}_0; \tag{3.26}$$

Objective function (3.1) is to minimize the total travelling cost as well as the total penalty cost for extra driving hours incurred from making the delivery for all collected orders from the port (L_0) to the final destinations. Constraint (3.2) is to ensure that all containers are delivered paired or individually. Constraint (3.3) forces containers to depart after they are ready to collect from the port, while constraint (3.4) means that all containers which are paired with another must depart after both are ready. Constraint (3.5) ensures that all containers that are paired depart at the same time. Constraints (3.6) and (3.7) guarantee that the gross weight of the whole vehicle which includes weights of the vehicle, chassis, containers and cargo do not exceed the maximum allowance. Constraints (3.8), (3.9) and (3.10) calculate the arrival times at containers' destinations, while constraints (3.11), (3.12) and (3.13) impose the time window restriction at the customer location. Constraints (3.14), (3.15), (3.16) and (3.17) calculate the total working time of the vehicle, by subtracting from the final arrival time at the empty leg destination the departure time from the port. Constraints (3.18) and (3.19) ensure that penalty is paid for extra working hours, while Constraints (3.20) and (3.21) restrict the model from planning routes that exceed the maximum working hours for one shift. Constraint (3.22) emphasises the fact that all orders with pre-determined empty leg destination must be delivered to the allocated location. Constraints (3.23) and (3.24) are there to ensure the total number of trucks used is no more than the corresponding fleet size. Finally, constraints (3.25) and (3.26) define the domains of the variables.

3.3 Practical Variant

Although the MILP model that is proposed in Section 3.2 is built to solve the pairing problem for import trips of containers only, variants of it, however, can be applied in numerous situations including both import, inland and/or export trips. Of course, the coverage of different situations is subject to necessary small modifications and appropriate interpretations of the parameters/data used. In this section we will discuss some major applications and how to adapt the model to achieve our aims.

3.3.1 Import (Export) of Containers Only

The scale of container business is different from one country to another, and very rarely a country has balanced import and export demands. In most developed countries like the UK, import is dominant; whereas in most developing countries such as China, export plays a far more important role than import. Although in general the pairing of import and export orders benefits, the imbalance in demands gives rise to potential decompositions of the entire delivery problem into subproblems having only import/export orders in it. Therefore, in the first part, we discuss around the original application for the combination of import orders only. To make this model more realistic, after all containers are served we consider an empty-leg trip to some inland depots where the empty containers are temporarily stored or a trip directly to an exporter that may have short term demands (but we do not consider the export trip explicitly). This makes a connection between the import and export trips which is normally used in practice. The model also allows customer specified empty-leg destination, which covers the case if a full container has already been allocated for an empty destination after it is delivered. Note that in our model we also allow a single container to be delivered to multiple (two) customer locations. This is seen in situations where customers are running relatively small business and when customers are sharing the cargo of a single container. Examples of combined routes are shown in Figure (3.1).

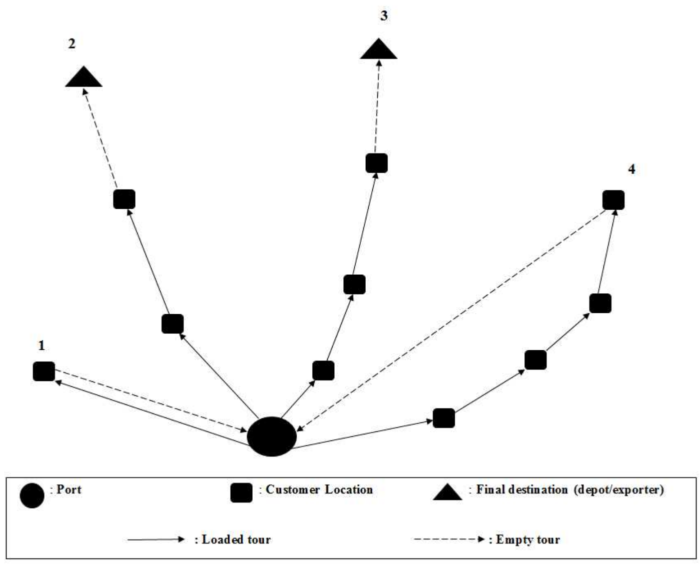

Figure 3.1: *Graphical illustration of application on the import of containers*

Based on the optimal decision of the model, there are four types of possible delivery routes as shown in Figure 3.1:

1. Deliver a container individually and then drop the empty container at an empty destination (port/inland depot/exporter).

2. Deliver two containers jointly, which both have a single destination and then drop the empty at an empty destination (port/inland depot/exporter).

3. Deliver two containers jointly, in which one has multiple destinations and then drop the empty at an empty destination (port/inland depot/exporter).

4. Deliver two containers jointly, which both have multiple destinations and then drop the empty at an empty destination (port/inland depot/exporter).

In contrast to import, the model can also be applied for the export-only case where empty containers should be picked up either from a depot/port or from an importer, travel for the pickup service and eventually deliver the loaded containers to the port.

3.3.2 Import (Export) and Inland Containers Transportation

In addition to the original problem setting, the MILP model can also be applied to the case where import trips are combined with inland trips. Container, as a means of safety delivery, is not only used in marine freight but also in inland transportation of bulk commodities. As traditionally the last-mile delivery of containers is carried out by haulage companies who also serve inland orders, the combination of import and inland trips are therefore vital in reducing unproductive travels. Note that in Subsections 3.3.2 and 3.3.3, unless stated in detail, we follow the problem statements and parameter descriptions proposed in Section 3.2. Here we consider two types of orders, each is associated with one type of container transportation requests, say the import orders and inland orders. An import order, like before, is a customer request of transporting a loaded container from the port to a customer location; an inland order, on the other hand, refers to the customer request of transporting one container's cargo from one inland location to another. Note that for the inland order, we assume that the customer does not own the container so that an empty container should be transported to the origin to do the loading, before visiting the destination for discharging. This defines the sequence following which the customer nodes should be visited for inland orders, which is in line with the full-twin assumption of multi-destination orders as mentioned before. So in this second scenario we take use of the multi-destinations order set $\mathcal{P}_2$ to assemble inland orders, and all decision variables relating to multi-destination orders are then interpreted as the "whether the inland order should be served by an individual trip or by a paired trip with another order. " In details, $x_{ijd} = 1, i \neq j \in \mathcal{P}_2$ means the inland order i should be combined with inland order j, so a 20ft truck collects an empty container from the port/inland depot, picks up order i's cargo from L_i^1, delivers it to L_i^2, then uses the same empty container to collect order j's cargo at L_j^1, delivers it to L_j^2 and finally drops the empty container to a nearby depot; $x_{ijd} = 1, i \in \mathcal{P}_1, j \in \mathcal{P}_2$ means the import order i should be combined with inland order j, so a 20ft truck picks up the loaded container i (an import order) from the port, delivers it to L_i, then uses the same empty container to collect order j's cargo at L_j^1, delivers it to L_j^2 and then drops the empty container to a nearby depot; $x_{ijd} = 1, i \neq j \in \mathcal{P}_1$ means the import order i is paired with another import order j, whereas $y_{id} = 1, i \in \mathcal{P}_2$ ($y_{id} = 1, i \in \mathcal{P}_1$) means the inland (import) order i is served individually.

Note that there are no longer import orders with multiple destinations as all orders in $\mathcal{P}_2$ are now interpreted as inland orders, therefore the usage of 40ft trucks in this case is only needed when two import orders are paired ($x_{ijd} = 1, i \neq j \in \mathcal{P}_1$). Constraints (3.23) and (3.24) are modified to:

$$\sum_{i\in\mathcal{N}}\sum_{d} y_{id} + \sum_{i\in\mathcal{N}}\sum_{j\in\mathcal{P}_2}\sum_{d} x_{ijd} \leq H_1,$$

$$\sum_{i\in\mathcal{P}_1}\sum_{j\in\mathcal{P}_1}\sum_{d} x_{ijd} \leq H_2$$

Weight constraints (3.6) and (3.7) should be modified accordingly as not in all paired cases we have two containers on the truck simultaneously.

$$V + W_i \sum_{d\in\mathcal{M}_0} (y_{id} + x_{ijd}) \leq V_1^m, \forall i \in \mathcal{N}, j \in \mathcal{P}_2$$

$$V + (W_i + W_j) \sum_{d\in\mathcal{M}_0} x_{ijd} \leq V_2^m, \forall i \neq j \in \mathcal{P}_1$$

Also for inland orders we need to set their "available time from the port", A_i, as zero (start of the day) so that it would not affect the departure time of the other container if it is paired, and use the time window to reflect its earliest available time at its pick up location. All other constraints stay the same as in the initial model.

Figure 3.2 shows graphically the possible delivery routes in optimal solution. In detail, they are:

1. An import order which is followed by an inland order.
2. An import order which is followed by an inland journey starting directly from the importer.
3. A single import order delivered individually.

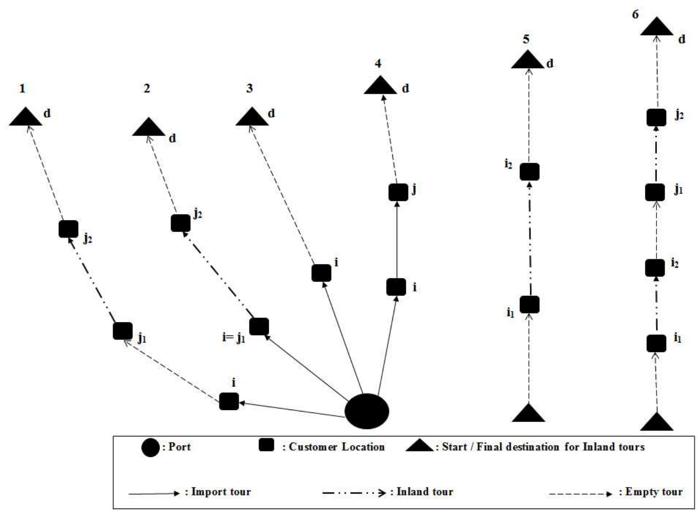

Figure 3.2: *Graphical illustration for the import-inland transportation*

4. Two import orders are paired on a 40ft truck and delivered one after another.

5. A single inland order delivered individually.

6. Two inland orders are paired and served one after another.

The same structure can also be used for the inland-export case, in which loaded containers are delivered from exporters to the port.

3.3.3 Import, Inland and Export Containers Transportation

Finally, we show that with small adaptations, our model can also be applied to the combination of import, inland and export trips. This is the most widely studied variant in existing literature which is believed very useful in reducing empty travels of containers by constructing a closed tour starting and ending at the port. Like the variant above, for this case we just need to adjust some interpretations of the model parameters, but a major improvement can be seen in the size of problems that can be solved exactly. As proposed in Subsection 3.3.2, we keep using the “multi-destination” subgroup $\mathcal{P}_2$ to capture inland orders. An export order, however, is defined as a customer request of transporting one container's cargo from a customer location to the port. As the destination for an export order is fixed at the port, we only need to know the origin, which can be any customer location, plus the standard parameters such as time windows and the weight of cargo to complete the definition. So an export order can be represented by a container d with single “destination” (which should be interpretated as origin here), $d \in \mathcal{P}_1$. Numerical examples of export orders are given in Table 3.5 as containers 6, 7 and 8. In this section we consider all orders types, say import, inland and export orders, each is associated with the transportation of a single container's cargo. Containers, which can be reused, are bound with chassis and to be filled in/stripped at customer locations. Empty containers are generated after the delivery of import/inland orders and are demanded before the pick up of inland/export orders. The problem is to find out how to make the transportation of all orders, individually or pairwisely, to achieve a minimum cost delivery plan satisfying the time, weight and working hours restrictions. Note that we do not consider the usage of 40ft trucks but only the combination of different types of trips on 20ft ones, because we define the variable only in a way that the empty container can be reused.

Observe that if we construct a complete return route with an import, an inland and an export order, the empty container is kept reused for the next task so that the empty leg destination is no longer needed. Therefore in the model we are going to use the previous “empty destination” d as the index for export orders, namely from L_d we load an empty container with cargoes to be delivered to the port. $x_{ijd} = 1, i \neq d \in \mathcal{P}_1, j \in \mathcal{P}_2$ is then interpreted as a 20ft truck collects a loaded container i from the port, delivers its cargo to L_i, then reuses the empty container to serve an inland order from L_j^1 to L_j^2, after which the same container is used to pick up cargo from L_d and delivers to the port. Note that in this case the number of variables actually reduces, since the round trip is only allowed in one way: import then inland then export. While to tackle with imbalances in these three types of demands we also allow individual trips for every type of request and the combined trips for every two types of request. So the decision variables are $x_{ijd}, i \neq d \in \mathcal{P}_1, j \in \mathcal{P}_2$ for combined trips with import, inland and export orders, $x_{ijd}, i \in \mathcal{P}_1, j \in \mathcal{P}_2, d \in \mathcal{M}_0$ for combined trips with import and inland orders, $x_{ijd}, i \neq j \in \mathcal{P}_1, d = port$ for combined trips with import and export orders, $y_{id}, i \in \mathcal{P}_2, d \in \mathcal{P}_1$ for combined trips with inland and export orders, $y_{id}, i \in \mathcal{P}_1, d \in \mathcal{M}_0$ for import trips, $y_{id}, i \in \mathcal{P}_2, d \in \mathcal{M}_0$ for inland trips and $y_{id}, i \in \mathcal{P}_1, d = port$ for export trips. Note that when d is not taken from $\mathcal{M}_0$, an additional term (L_0) should be added to the travel distance and travel time function to include the travelling from L_d to the port. Also the fleet size constraints (3.23) and (3.24) should be combined into one as H_2 (40ft trucks) no longer exists:

$$\sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{N}} \sum_{d} x_{ijd} + \sum_{i \in \mathcal{N}} \sum_{d} y_{id} \leq H_1,$$

as well as the weight constraints (3.6) and (3.7):

$$V + W_i \sum_{d} (y_{id} + x_{ijd} + x_{jid}) \leq V_1^m, \forall i \in \mathcal{N}$$

As shown in Figure 3.3 and as we explained above, there will be different possibilities for this case which are:

1. A loaded order as import, followed by an inland order and finally the export order is

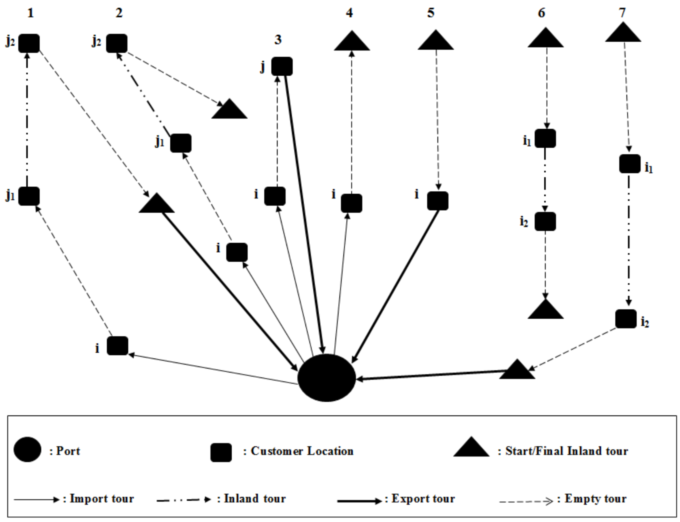

Figure 3.3: *Graph explains the import-inland-export transportation*

delivered to the port.

2. A loaded order as import, followed by an inland order and the empty container is delivered to an empty storage (port/inland depot/exporter).

3. A loaded order as import followed by an export tour.

4. A single import order served individually.

5. A single export order served individually.

6. A single inland order served individually.

7. Empty container is picked up from a nearby empty storage (port/inland depot/exporter) to start a single inland tour, which is then followed by an export trip.

3.4 Numerical examples

In this section we construct small examples according to the three applications as discussed in Section 3.3, and test our model against them to show the performance. The MILP model is coded in MPL and solved by Cplex.

3.4.1 Example 1 – Import Only

In this example, 5 containers are to be delivered from a single port to a subgroup of 10 customer locations, (1, ..., 7), where (0, 8, 9, 10) are defined as port/depot/exporter. Data is summarised in Table 3.1. For instance container 1, whose cargo weighs 8,900 (kg), is available to pick up from the port at time 6.30 a.m and should be delivered to customer location 1 between 8am and 2pm.

Containers	L_i	Ai	$[T_i^s, T_i^e]$	$W_i(kg)$
1	1	6.30 a.m	[8.00, 14.00]	8900
2	2	6.00 a.m	[7.00, 16.00]	12900
3	3	7.00 a.m	[9.00, 17.00]	22900
4	4	8.00 a.m	[10.00, 16.00]	10900
4	5	8.00 a.m	[10.00, 20.00]	10900
5	6	9.00 a.m	[8.00, 18.00]	13900
5	7	9.00 a.m	[9.00, 19.00]	13900

Table 3.1: *Data for import-only example*

Containers 1, 2, 3 are for single destination while containers 4 and 5 are allocated to two destinations each. In addition, there are 8 lorries available for the import service, with 4 for paired delivery and 4 for single delivery. For all trucks a gross weight limitation of $44,000$kg is applied, which includes the truck ($7,500$kg), the chassis ($4,800$kg), the empty container ($2,300$kg) and the cargo (given in the table as W_i). Maximum working time of

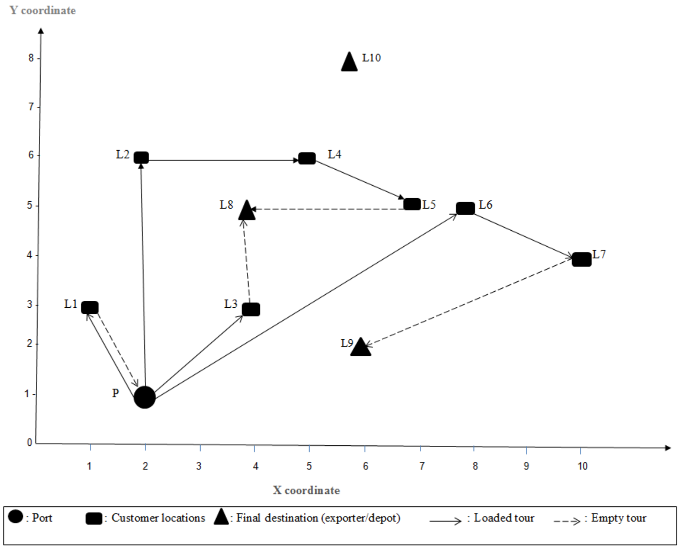

Figure 3.4: *Solution for example 1 (import of containers)*

Import containers	Route sequence	Departure time(port)	First importer arrival time	Last Importer arrival time	Penalty cost	Final destination
2,4	L_2, L_4, L_5	10.00	L_2: 12.50	L_5: 19.11	30	8
1	L_1	12.50	L_1: 14.00	-	0	0
3	L_3	11.00	L_3: 12.41	-	0	8
5	L_6, L_7	10.50	L_6: 14.10	L_7: 17.22	0	9

Table 3.2: *Solution for example 1 (import of containers)*

truck driver is set to 9 hours (according to UK regulation) and a penalty of 30 pounds should be paid for any extra hours driven. Average service time at all customer locations is 2 hours. The empty containers should be delivered to the depot/port/exporter which minimizes the total travelling distance.

The solution to this example is displayed in Figure 3.4. In the solution there are four trips, one travelled by a 40ft long lorry to carry containers 2 and 4 pair-wisely, three travelled by 20ft long lorries to deliver containers 1, 3 and 5 individually. Dashed lines represent the empty container movement. Table 3.2 gives more detailed information about the departure and arrival times at all customer locations. The result makes perfect sense. First, container 3 cannot be delivered pair-wisely as its weight is too high to combine with any other container. Second, it is impossible to combine containers 4 and 5 (both for multi-destination) due to the time window constraints. Thirdly, although we need to pay for the overdue in working time for the paired delivery route, the total distance travelled is largely reduced compared with delivering containers 2 and 4 individually. Given these observations, the solution displayed in Figure 3.4 is optimal.

3.4.2 Example 2 – Import-Inland Transportation

Now we implemented the MILP for the import-inland transportation. As given in Table 3.3, there are 6 shipping requests of containers under consideration and two of them (container 4 and 6) are inland requests. Here we assume that the time window applies to the origin and destination nodes, and the time window for the port are already existing in the model as the available time to pick up containers. The weights of containers are also specified.

Containers	Origin	$[T_i^s, T_i^e]$	Destination	$[T_i^s, T_i^e]$	$W_i(kg)$
1	port	-	L_1	[8.00, 14.00]	8900
2	port	-	L_2	[7.00, 16.00]	12900
3	port	-	L_3	[9.00, 17.00]	12900
4	L_4	[9.00, 16.00]	L_5	[10.00, 20.00]	10900
5	port	-	L_6	[8.00, 18.00]	13900
6	L_6	[8.00, 18.00]	L_7	[9.00, 19.00]	11900

Table 3.3: *Data for import-inland example*

paired orders	Departure time(port)	Import orders arrival time	Inland order arrival time	Final destination (port/exporter/depot)
1,2	10.00	L_1:11.11 , L_2: 14.40	-	8
3,4	11.00	L_3:12.43	L_4: 19.11	8
5,6	10.50	L_6: 14.10	L_6: 14.10	9

Table 3.4: *Solution for example 2 (import-inland transportation)*

Solving this example by the MILP model, it creates paired delivery routes for all con-

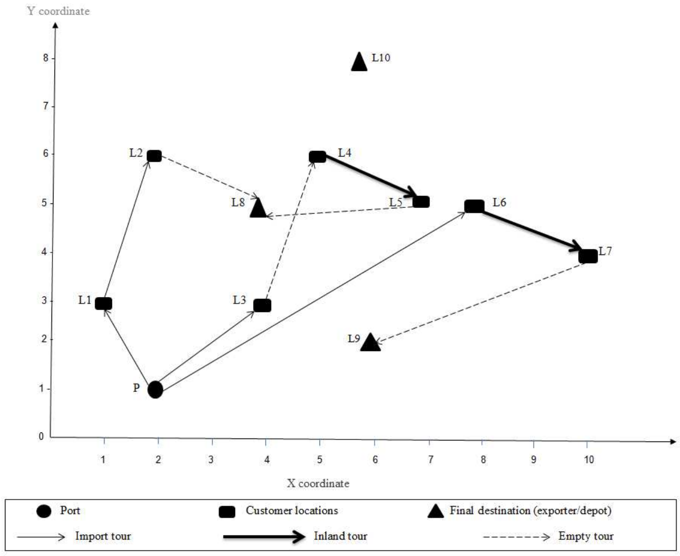

Figure 3.5: *Solution for example 2 (import and inland delivery)*

tainers like shown in Table 3.4 and Figure 3.5. Specifically, two import orders, containers 1 and 2 are paired to form a trip that departs from the port at 10am and finishes at depot 8; import container 3 and inland container 4 are paired to form a trip that departs from the port at 11am and finishes at depot 8; import container 5 and inland container 6 are paired to form a trip that departs from the port at 10.50am and finishes at depot 9.

3.4.3 Example 3 – Import-Inland-Export transportation

In this example we tested the model for combining the import, inland and export orders together. As seen in Table 3.5, 8 orders are considered to deliver from their origins to final destinations, as well as the time windows and weight of each request of containers.

Looking at the results given in Table 3.6 and Figure 3.6 we can see, in this case the code has paired orders 1, 3 and 6, so the 20ft long lorry carries a single loaded container (container 1) and departs from the port at 11.00am to visit its allocated importer, location 1, at which the inland trip is started towards location 3, then picks up an export loaded

Containers	Origin	$[T_i^s, T_i^e]$	Destination	$[T_i^s, T_i^e]$	$W_i(kg)$
1	port	-	L_1	[8.00, 14.00]	8900
2	port	-	L_2	[7.00, 16.00]	12900
3	L_1	[8.00, 14.00]	L_3	[9.00, 17.00]	12900
4	L_4	[9.00, 16.00]	L_5	[10.00, 16.00]	10900
5	port	-	L_6	[8.00, 18.00]	13900
6	L_3	[8.00, 20.00]	port	-	11900
7	L_5	[10.00, 21.00]	port	-	14900
8	L_7	[9.00, 19.00]	port	-	13900

Table 3.5: *Data for import-inland-export example*

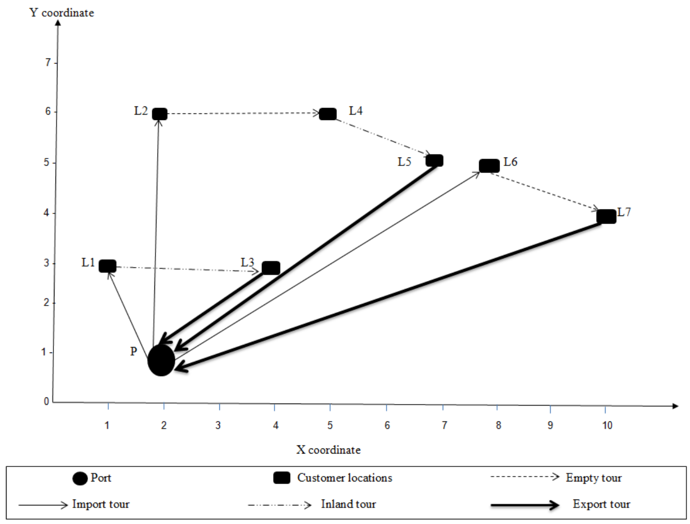

Figure 3.6: *Solution for example 3 (import-inland-export transportation)*

container from location 3 to deliver to the port. On the other hand, import order 2 is combined with inland order 4 and export order 7. This means a 20ft long lorry leaves from the port at 10.00am carrying container 2, discharges it at location L_2 at 12.50pm and the empty truck is then moved to location 4 to start the inland tour from L_4 to L_5. The export tour starts at 19.11 from location 5 to the port. Similarly, orders 5 and 8 are paired on a 20ft long lorry travelling from the port at 12.27pm and ending its return trip at the port after picking up the export order from L_7 at 19.00. Penalty cost is charged on route (2, 4, 7), which violates the maximum working time regulation.

paired Orders	departure time(port)	import tours	inland tours	export tours	import tour arrival time	inland tour arrival time	export tour arrival time
1,3,6	11.00	$L_0 - L_1$	$L_1 - L_3$	$L_3 - L_0$	L_1:12.11	-	L_3:15.37
2,4,7	10.00	$L_0 - L_2$	$L_4 - L_5$	$L_5 - L_0$	L_2:12.50	L_4:16.00	L_5:19.11
5,8	12.27	$L_0 - L_6$	-	$L_7 - L_0$	L_6:15.37	-	L_7:19.00

Table 3.6: *Solution for example 3 (import-inland-export transportation)*

3.4.4 Real Implementations

To test the performance of the MILP model for real life instances, geographical information of the Port of Felixstowe, which is one of the major ports in the UK, and its major service areas are considered. Orders are represented by the number of 20ft containers that should be distributed from the port to inland customer locations, between a pair of inland locations and from exporters/inland depots to the port. As for the convenience and diversity of tests, apart from the geographical location all other data is randomly generated. Instance sizes range from 10 to 300 orders per day are considered, which meets the basic service level of a medium sized haulage company. Distances are calculated based on the straight line distance which is rescaled by 1.3 as an approximation to the road-distance. The average speed for lorries is randomly picked within [35, 40] mile/h and the penalty cost is 200 pounds/h for extra working hours. Loading containers at the port takes no time (h=0) as it has been considered in the container available time, while at customer locations it takes about 2 hours. As mentioned above the model is coded with MPL and solved by Cplex, on a CPU with an Intel(R)Core(TM)i7-4790 processor. In what follows we will show the numerical results of testing the model against three types of applications.

As it can be seen in Table 3.7, it explains the result for the import of containers. In each instance we consider different numbers of loaded orders (containers), some have a single destination while some others have dual destinations. More than 300 locations distributed

# Orders	# Containers		#Importers	#Depots	# Indv. fleet		# Paired. fleet		Cost			CPU time(sec)
	# single dest.	# multi dest.			avail.	used	avail.	used	indv.	paired	Penalty	
10	5	5	16	4	5	4	5	3	67.3	381.2	200	00:03
	6	4	15	5	5	4	5	3	67.1	210	-	00:02
	7	3	14	6	5	4	5	3	82.5	242.1	-	00:02
50	44	6	44	7	25	24	20	13	292.5	568.8	-	08:72
	43	7	43	8	15	14	25	18	192.8	1011.3	600	07:41
	40	10	42	9	25	24	15	13	446.6	737.1	-	11:79
100	88	12	91	10	40	40	40	30	596.6	1512.6	200	114:00
	90	10	90	11	45	44	45	28	609.3	1378.8	-	89:00
	80	20	81	20	45	44	35	28	784.8	1227.9	-	86:00
150	135	15	128	23	70	70	70	40	1081.9	1567.4	-	97:00
	110	40	136	15	60	60	60	45	973.4	2467.9	-	124:00
	115	35	131	20	60	60	45	56	926.2	2431.3	-	174:00
200	175	25	185	16	70	70	70	65	1134.6	3535.1	200	463:00
	180	20	183	18	100	100	80	50	1580.2	2267.5	-	270:04
	185	15	180	21	60	60	75	70	914.6	3750.9	-	223:00
250	230	20	234	17	130	130	100	60	2218.6	2800.5	400	600:04
	220	30	230	21	110	110	90	70	1340.7	2441.7	-	561:00
	210	40	226	25	80	80	90	85	1096.2	4025.6	-	879:03
300	265	35	275	26	100	100	110	100	3051.4	7959.6	-	1235:04
	260	40	280	21	150	150	130	75	2393.9	2353.9	-	1302:00
	255	45	270	31	140	140	120	80	2003.4	2144.6	-	1212:00
350	300	50	319	32	170	170	140	90	2730.6	3405.5	-	1920:00
	310	40	325	26	160	160	130	95	2811.9	3955.9	-	1838:00
	305	45	316	35	150	150	120	100	2418.2	3960	-	1922:00

Table 3.7: *Results for large size instances drawn from real geographical data – import only*

Orders		**Without pairing**			**With pairing**				**Cost gap (%)**
Import	Inland	# tours	O.F	CPU time(sec)	Indiv. tours	paired tours	O.F	CPU time(sec)	
5	5	10	634.1	00:03	6	2	532.9	00:03	16%
25	25	50	2225.5	00:54	20	15	1800.6	11:03	19%
50	50	100	5071.8	02:40	34	33	3976.4	86:00	22%
70	80	150	8162.6	07:98	88	31	7328.4	203:00	10%
130	70	200	10206	17:58	102	49	8232.7	215:00	19%
150	100	250	13926.1	26:75	46	102	9607.6	526:00	31%
250	50	300	12128.8	59:17	36	132	7379.2	1323:00	39%
300	50	350	14470	165:00	70	140	9140.4	1992:00	37%

Table 3.8: *Results for different real instances of the import-inland transportation*

Orders			**Without pairing**			**With pairing**					**Cost gap (%)**
Imp.	Inl.	Exp.	# tours	O.F	CPU time(sec)	Indiv. tours	paired 2 orders	paired 3 orders	O.F	CPU time(sec)	
4	5	6	15	952.7	00:01	3	6	-	855.8	00:03	10%
19	25	29	73	3450	00:10	10	24	5	2845	85:00	17%
31	49	55	135	5711.7	00:46	3	33	22	3182.1	119:00	44%
62	80	87	229	11341.8	00:98	34	66	21	9769.8	398:00	14%
107	75	93	275	14902.1	01:85	63	67	26	12150.1	1581:00	18%
121	100	129	350	16645.7	02:94	20	57	72	11407.8	1832:00	31%

Table 3.9: *Results for different real instances of the import-inland-export transportation*

around southeast England are considered as the number of customers (importers) where the loaded containers should be delivered to. We also consider different numbers of inland depots (where empty containers should be delivered to) across examples. A number of 20ft and 40ft long trucks are available to use, which can carry one or two 20ft containers respectively. It is clear from the results that in some cases a penalty should be paid for extra working hours for some planned routes. Looking at the results we can see, it is not always economical to use up the entire 40ft fleets. There are three main reasons for this observation: first, some containers are not able to be paired with others due to the weight restriction; second, the penalty paid for extra working hours of a paired trip might be higher than the extra distance travelled by sending two individual trucks, especially when many containers are nominated for multiple destinations; thirdly, the existence of inland depots makes the individual delivery less costly than doubling the total travel distance of the paired trip, as empty containers can be easily dropped at a nearby depot. A major notice should be put on the solution time of the model, as in all existing literature that are known by the authors, no one can solve this type of problem with 350 orders within about half an hour, not to mention after the inclusion of multi-destination orders which is introduced for the first time in this work. Based on the result we have the reason to believe, the assignment model as proposed does solve more efficiently than the VRP-SPDTW on the same type of container pairing problems. On the other hand, the result of the import-inland delivery is shown in Table 3.8, where a number of import orders that should be delivered from the port to their destinations and a number of inland orders that should be delivered from one inland location (origin) to another (destination). In order to see by how much the combined delivery can reduce transportation costs, the solution of the MILP is compared with a trivial solution where all orders are served individually by 20ft trucks. The result for this benchmark solution is given under "Without pairing" columns, whereas the solution of the MILP is displayed under "With pairing". Note that the "With pairing" case also allows individual delivery. The optimal decision is simply picked up by the MILP model minimizing the total working cost (transportation plus penalty). The result shows that the minimum cost (O.F) for the paired case is 10-39% less than the cost (O.F) for the individual delivery across all cases that we have tested. In general, when the inland orders take a high proportion in the overall orders pool, the improvement of pairing is less significant over individual delivery. This is due to the fact that, with existences of inland depots, finding a nearby depot to start/end the inland trip is not difficult. So the necessity of combining the trip with an import or another inland order is diluted. On the other hand, as there is only one port which is normally far away from customer locations, the combination of orders for import delivery is more vital in reducing transportation cost. This also justifies our initial argument that emphasises should be made to the pairwise delivery import/export orders only.

Table 3.9 shows the result for the combination of import-inland-export orders. Similar as the import-inland case, allowing combination of tours saves at most 44% of the total delivery cost. Notice that Table 3.9 gives the detailed number of tours that combined 1, 2 or 3 containers, these in turn represents the number of import/inland/export tours that are served individually, the number of import-inland/import-export/inland-export tours that are served pair-wisely, and the number of import-inland-export tours. The individual tour in this case is largely reduced with combination, which justifies the preference of using combined delivery as well.

3.5 Summary

In this chapter the delivery of 20ft orders (containers) from their origins to destinations is investigated, an assignment MILP optimization model is formulated for the Pairing of Containers/Orders in Drayage Transportation (PCDT) with the aim of minimizing the travelling cost and penalty paid for over time working. A great number of realistic restrictions are considered in the model such as time windows at customer locations, and working time regulations, ready time of containers at the port, the usage of inland depots to reduce empty travels, etc. In addition, this work also allows containers to be delivered to multi destinations for discharging, which is economically convenient for customers running relatively small business. The model can be implemented for different types of transportation such as the import (delivery) of containers, import-inland as well as the import-inland-export. The decision of delivering orders paired or individually can be made efficiently by solving the MILP model using commercial software like Cplex. Even under a dense inland depot setting, a 23% operations cost reduction is achievable on average across all testing examples. Testing on numerical examples drawn from realistic geographical data shows that up to 350 orders can be solved using the MILP model within a reasonable time (about 30 minutes), which outperforms traditional models that are based on the VRP-SPDTW which normally solves instances up to 75 Vidovic et al. [99]. Without needing any heuristics, more accurate and reliable solution can be achieved efficiently by the proposed model.

Chapter 4

Heterogenous Container Sizes

4.1 Introduction

The inland container transportation can be classified as a pickup and delivery problem. Parragh et al. [78] provided an extensive classification and explanation of the pickup and delivery of vehicle routing problems. However, inland container transportation is an extension of these types since it comprises the pickup and delivery of different types of containers (loaded and empty) between customers, the port and inland depots [67]. Falling in the category of vehicle routing problem, this problem can be classified as a vehicle routing problem with pickup and delivery (VRPPD) or as a vehicle routing problem with backhauls (VRPB). The PDVRP of containers can be described where trucks are able to visit import and export customers in any feasible sequence, while for the VRPB, all import customers should be served before export customers.

In this chapter, we expand Chapter 3 by considering both heterogeneous truck types and container sizes (20ft and 40ft). We design a Mixed Integer Linear Programming (MILP) model for the combination of orders in the inland transportation of containers by truck. In this model, the pickup and delivery of both 20 and 40 foot containers from the terminal to customers locations and vise versa are considered. We assume that all customers use the *Stripe* strategies, which means containers are loaded onto/removed from the truck together with their contents. Empty movements are treated as separate requests, with zero payload weight. The model is an extended version of our previous work, which considered only 20ft, fully loaded shipments. All other practical restrictions remain, such as the time window of the required work for both the port and all customer locations, the weight restrictions for single and double chassis types, the penalty cost for the potential overtime working of truck drivers, etc. Based on the same assignment structure as in Chapter 3, this model solves efficiently the problem in question: examples with up to 100 orders can be solved by CPLEX, which generates the optimal delivery plan satisfying all constraints.

In order to deal with larger instances, a decomposition and aggregation heuristic is designed. The basic idea of this approach is to decompose the locations of orders geographically into small subgroups and solve the subgroups problems by the formulated MILP, to obtain the optimal order-combination plan within each subproblem. For the sake of constructing more combined routes which save delivery cost in general, decomposition is proposed based on the angles of the customer location to the port, which creates fan-shaped subareas for each decomposed order group. To balance fleet sizes amongst all subgroups and to achieve the best heuristic optimal solution, column generation is used to iteratively adjust the number of allocated trucks according to the shadow-price of each truck type. The decomposed model is smaller in size and solves efficiently, but the quality of the result is not highly optimal since we prevent some combination choices through decomposition. Therefore in the second phase, we do aggregation by removing the "best" combined orders, e.g. the order combinations that have already used the full truck load, in order to reduce the problem size. A new MILP is constructed for the aggregated data, and solved as usual. This decomposition and aggregation approach is tested to be both efficient and cost-saving through intensive numerical experiments.

This chapter is structured as follows. In Section 4.2 the problem description and the optimisation model are described. The heuristic approach is demonstrated with an example in Section 4.3. Numerical experiments are presented in Section 4.4. Section 4.5 draws the summary.

4.2 Problem Description and Optimization Model

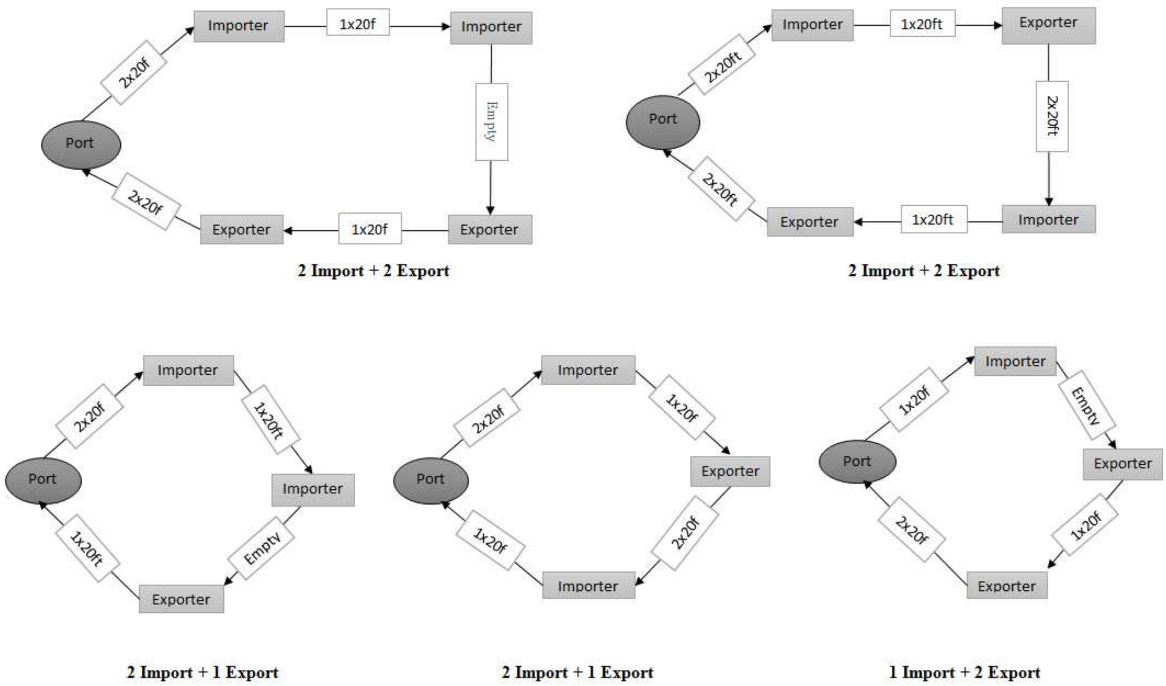

Figure 4.1: *Combination Possibilities of import/export 20/40ft orders*

In order to describe the problem, the basic terminology of this chapter will be defined in this section. The transportation of a customer's request of a 20ft or 40ft container from its origin to its destination is referred to as an *order*. The process of transporting a container from the port to the customer location is called an *import* order. In contrast, the collection of a container from the customer location to the port is defined as an *export* order. Figures 4.1, 4.2 and 4.3 illustrate the different possibilities of combining orders of the problem.

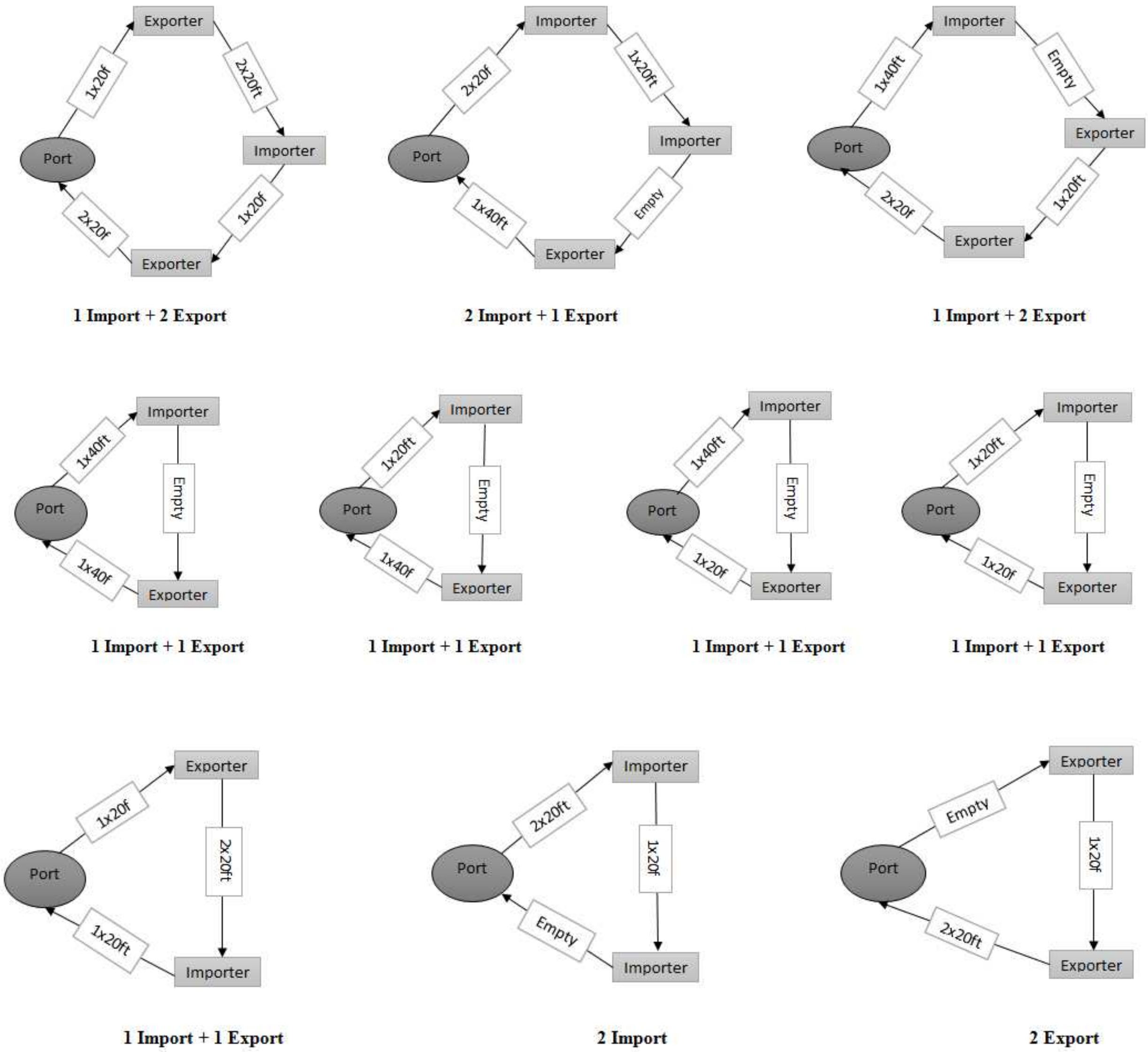

Figure 4.2: *Combination Possibilities of import/export 20/40ft orders*

It is assumed that a haulage company has a number of heterogenous trucks in its fleet: 20ft and 40ft trucks/chassises that are available for the service. The import and export of orders (20ft and 40ft containers) should be collected by the company from the order origin and delivered to the order destination. It is assumed that the whole container is left at the customer's location at the end of the service, and the empty truck continues its route to fulfil other orders. Each order is allocated for a single customer location, has a specific time window for delivery/pick up and a payload weight of the container, including the cargo. Furthermore, we assume all information and data of the problem are known in advance. Based on these assumptions, a decision should be made on which orders should be combined to form a route as shown in Figures 4.1, 4.2 and 4.3, to minimize the total traveling cost and the penalty cost of violating the working time regulation for truck drivers.

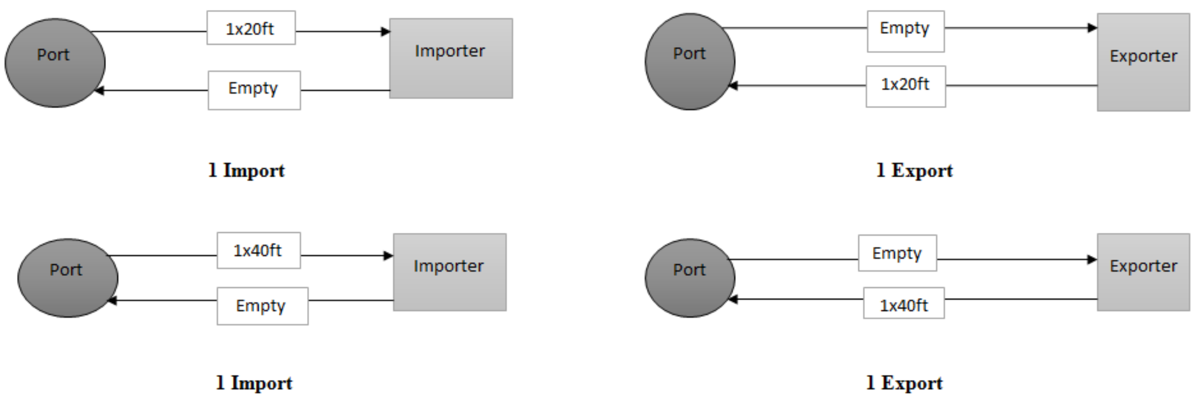

Figure 4.3: *Individual delivery of import/export 20/40ft orders*

4.2.1 Parameters

In this section we illustrate the parameters of the model. Let H_1 and H_2 denote the number of 20ft and 40ft trucks that are available during the day, respectively. Let $\mathcal{N}$ be a set of containers which consists of: $\mathcal{P}_1$ as a set of 20ft import orders, $\mathcal{P}_2$ as a set of 20ft export orders, $\mathcal{P}_3$ as a set of 40ft import orders and $\mathcal{P}_4$ as a set of 40ft export orders. Note that here we represent the needs of taking a container from the port to a customer location as an "import" order, regardless of whether this container is a real "import" (loaded) order or a request to re-locate an empty container for the following export services. This means in this study, we ignore the fact that some containers might be loaded and some others are empty, and simply describe an "order" as a customer request of delivering a container from its origin to its destination. We are not concerned about how the empty container will be processed and re-located after fulfilling the current delivery request. If there is such a need, a new order could be generated to capture it. Orders should be picked up from their origins, either from the port (L_o) for import orders or from a customer location ($L_i, i \in \mathcal{N}$) for export orders with a specific available time (A_i) and a handling time (h). The service time for orders to be striped at the customer's location is denoted as (O). The travel time $t(.,...,.)$ for a sequence of locations is restricted by the driving hours regulation, where the maximum possible working time T_{max} applies for a single shift. If it is violated, a penalty cost C should be paid for per unit of the extra working time. Each order has an earliest and latest time window $[T_i^s, T_i^e]$ to arrive at its destination. The payload weight of containers is defined as ($W_i, i \in \mathcal{N}$), while the weight of the truck chassis is denoted by (V). Note that for an order with an empty container, the payload weight W_i is zero. When delivered jointly, the weight limit V_2^m for the whole truck is considered. From the perspective of haulage companies, the delivery cost $f(.,...,.)$ is estimated based on the distance between locations.

4.2.2 Decision Variables

In order to capture the characteristics of the optimization model, two types of decision variables are introduced. Binary variables $x_{ijkl}, x_{ijk}, x_{ij}$ and x_i are created to denote the decision on how the orders should be combined for delivery. For example, $x_{ijkl} = 1$ means orders i, j, k and l should be delivered on the same route, following the sequence of i to j to k and then to l. Another set of binary variables $y_{ijkl}, y_{ijk}, y_{ij}$ and y_i are defined to identify if a penalty cost should be paid for the associated delivery route for extra working hours. Moreover, continuous variables v_i are introduced to compute the arrival times for order i at its receiver.

Binary Variables

- $x_{ijkl} = \begin{cases} 1, & \text{if containers } i, j, k \text{ and } l \text{ are delivered on the same route,} \\ & \text{following the sequence of } i, j, k \text{ and then } l. \\ 0, & \text{otherwise} \end{cases}$
  $\forall i \neq j \in \mathcal{P}_1, \forall k \neq l \in \mathcal{P}_2 \;\; or \;\; \forall i \neq k \in \mathcal{P}_1, \forall j \neq l \in \mathcal{P}_2.$
- $x_{ijk} = \begin{cases} 1, & \text{if containers } i, j \text{ and } k \text{ are delivered on the same route,} \\ & \text{following the sequence of } i, j \text{ and then } k. \\ 0, & \text{otherwise} \end{cases}$
  $\forall i \neq j \in \mathcal{P}_1, \forall k \in \mathcal{P}_2 \cup P_4 \;\; or \;\; \forall i \in \mathcal{P}_1 \cup P_3, \forall j \neq k \in \mathcal{P}_2$

  $or \;\; \forall i \neq k \in \mathcal{P}_1, \forall j \in \mathcal{P}_2 \;\; or \;\; \forall i \neq k \in \mathcal{P}_2, \forall j \in \mathcal{P}_1.$
- $x_{ij} = \begin{cases} 1, & \text{if containers } i \text{ and } j \text{ are delivered on the same route,} \\ & \text{following the sequence of } i \text{ and then } j. \\ 0, & \text{otherwise} \end{cases}$
  $\forall i \in \mathcal{P}_1 \cup P_3, \forall j \in \mathcal{P}_2 \cup P_4 \;\; or \;\; \forall i \neq j \in \mathcal{P}_1$

  $or \;\; \forall i \neq j \in \mathcal{P}_2 \;\; or \;\; \forall i \in \mathcal{P}_2, \forall j \in \mathcal{P}_1.$

- $x_i = \begin{cases} 1, & \text{if container } i \text{ is delivered individually to its destination.} \\ 0, & \text{otherwise} \end{cases}$

  $\forall i \in \mathcal{P}_1 \cup P_2 \cup P_3 \cup P_4$.

- $y_{ijkl} = \begin{cases} 1, & \text{if working hours for the combined route } x_{ijkl} \text{ is higher than the} \\ & \text{maximum regulation working time.} \\ 0, & \text{otherwise} \end{cases}$

  $\forall i \neq j \in \mathcal{P}_1, \forall k \neq l \in \mathcal{P}_2 \quad or \quad \forall i \neq k \in \mathcal{P}_1, \forall j \neq l \in \mathcal{P}_2$.

- $y_{ijk} = \begin{cases} 1, & \text{if working hours for the combined route } x_{ijk} \text{ is higher than the} \\ & \text{maximum regulation working time.} \\ 0, & \text{otherwise} \end{cases}$

  $\forall i \neq j \in \mathcal{P}_1, \forall k \in \mathcal{P}_2 \cup P_4 \;\; or \;\; \forall i \in \mathcal{P}_1 \cup P_3, \forall j \neq k \in \mathcal{P}_2$

  $or \;\; \forall i \neq k \in \mathcal{P}_1, \forall j \in \mathcal{P}_2 \;\; or \;\; \forall i \neq k \in \mathcal{P}_2, \forall j \in \mathcal{P}_1$.

- $y_{ij} = \begin{cases} 1, & \text{if working hours for the combined route } x_{ij} \text{ is higher than the} \\ & \text{maximum regulation working time.} \\ 0, & \text{otherwise} \end{cases}$

  $\forall i \in \mathcal{P}_1 \cup P_3, \forall j \in \mathcal{P}_2 \cup P_4 \;\; or \;\; \forall i \neq j \in \mathcal{P}_1$

  $or \;\; \forall i \neq j \in \mathcal{P}_2 \;\; or \;\; \forall i \in \mathcal{P}_2, \forall j \in \mathcal{P}_1$.

- $y_i = \begin{cases} 1, & \text{if working hours for the individual route } x_i \text{ is higher than the} \\ & \text{maximum regulation working time.} \\ 0, & \text{otherwise} \end{cases}$

  $\forall i \in \mathcal{P}_1 \cup P_2 \cup P_3 \cup P_4$.

Continuous Variables

- $v_i, \forall i \in \mathcal{P}_1 \cup P_2 \cup P_3 \cup P_4$: arrival time for order i at its destination.

4.2.3 Mathematical Model

A Mixed Integer Linear Programming (MILP) model is formulated to find the best combination decisions for delivery route planning. The model consists of the most important practical constraints that are normally used in industry.

$$
\begin{aligned}
\min \sum_{i\in\mathcal{P}_1}\sum_{j\in\mathcal{P}_1}\sum_{k\in\mathcal{P}_2}\sum_{l\in\mathcal{P}_2}&(f(L_o,L_i,L_j,L_k,L_l,L_o)x_{ijkl}+f(L_o,L_i,L_k,L_j,L_l,L_o)x_{ikjl})+\\
&\sum_{i\in\mathcal{P}_1}\sum_{j\in\mathcal{P}_1}\sum_{k\in\mathcal{P}_2\cup\mathcal{P}_4} f(L_o,L_i,L_j,L_k,L_o)x_{ijk}+\sum_{i\in\mathcal{P}_1\cup\mathcal{P}_3}\sum_{k\in\mathcal{P}_2}\sum_{l\in\mathcal{P}_2} f(L_o,L_i,L_k,L_l,L_o)x_{ikl}+\\
&\sum_{i\in\mathcal{P}_1}\sum_{k\in\mathcal{P}_2}\sum_{j\in\mathcal{P}_1} f(L_o,L_i,L_k,L_j,L_o)x_{ikj}+\sum_{k\in\mathcal{P}_2}\sum_{i\in\mathcal{P}_1}\sum_{l\in\mathcal{P}_2} f(L_o,L_k,L_i,L_l,L_o)x_{kil}+\\
&\sum_{i\in\mathcal{P}_1\cup\mathcal{P}_3}\sum_{k\in\mathcal{P}_2\cup\mathcal{P}_4} f(L_o,L_i,L_k,L_o)x_{ik}+\sum_{i\in\mathcal{P}_1}\sum_{j\in\mathcal{P}_1} f(L_o,L_i,L_j,L_o)x_{ij}+\\
&\sum_{k\in\mathcal{P}_2}\sum_{i\in\mathcal{P}_1} f(L_o,L_k,L_i,L_o)x_{ki}+\sum_{k\in\mathcal{P}_2}\sum_{l\in\mathcal{P}_2} f(L_o,L_k,L_l,L_o)x_{kl}+\sum_{i\in\mathcal{P}_1\cup\mathcal{P}_2\cup\mathcal{P}_3\cup\mathcal{P}_4} f(L_o,L_i,L_o)x_i+\\
&\sum_{i\in\mathcal{P}_1}\sum_{j\in\mathcal{P}_1}\sum_{k\in\mathcal{P}_2}\sum_{l\in\mathcal{P}_2}(Cy_{ijkl}+Cy_{ikjl})+\sum_{i\in\mathcal{P}_1}\sum_{j\in\mathcal{P}_1}\sum_{k\in\mathcal{P}_2\cup\mathcal{P}_4} Cy_{ijk}+\sum_{i\in\mathcal{P}_1\cup\mathcal{P}_3}\sum_{k\in\mathcal{P}_2}\sum_{l\in\mathcal{P}_2} Cy_{ikl}\\
&+\sum_{i\in\mathcal{P}_1}\sum_{k\in\mathcal{P}_2}\sum_{j\in\mathcal{P}_1} Cy_{ikj}+\sum_{k\in\mathcal{P}_2}\sum_{i\in\mathcal{P}_1}\sum_{l\in\mathcal{P}_2} Cy_{kil}+\sum_{i\in\mathcal{P}_1\cup\mathcal{P}_3}\sum_{k\in\mathcal{P}_2\cup\mathcal{P}_4} Cy_{ik}+\sum_{i\in\mathcal{P}_1}\sum_{j\in\mathcal{P}_1} Cy_{ij}+\\
&\sum_{k\in\mathcal{P}_2}\sum_{i\in\mathcal{P}_1} Cy_{ki}+\sum_{k\in\mathcal{P}_2}\sum_{l\in\mathcal{P}_2} Cy_{kl}+\sum_{i\in\mathcal{P}_1\cup\mathcal{P}_2\cup\mathcal{P}_3\cup\mathcal{P}_4} Cy_i;
\end{aligned}
\tag{4.1}
$$

s.t.

$$
\begin{aligned}
&\sum_{j\in\mathcal{P}_1}\sum_{k\in\mathcal{P}_2}\sum_{l\in\mathcal{P}_2}(x_{ijkl}+x_{jikl}+x_{ikjl}+x_{jkil})+\sum_{j\in\mathcal{P}_1}\sum_{k\in\mathcal{P}_2\cup\mathcal{P}_4}(x_{ijk}+x_{jik})+\sum_{k\in\mathcal{P}_2}\sum_{l\in\mathcal{P}_2}(x_{ikl}+x_{kil})\\
&+\sum_{k\in\mathcal{P}_2}\sum_{j\in\mathcal{P}_1}(x_{ikj}+x_{jki})+\sum_{j\in\mathcal{P}_1}(x_{ij}+x_{ji})+\sum_{k\in\mathcal{P}_2}(x_{ik}+x_{ki})+\sum_{k\in\mathcal{P}_4} x_{ik}+x_i=1, \forall i\in\mathcal{P}_1;
\end{aligned}
\tag{4.2}
$$

$$
\begin{aligned}
&\sum_{i\in\mathcal{P}_1}\sum_{j\in\mathcal{P}_1}\sum_{l\in\mathcal{P}_2}(x_{ijkl}+x_{ijlk}+x_{ikjl}+x_{iljk})+\sum_{i\in\mathcal{P}_1\cup\mathcal{P}_3}\sum_{l\in\mathcal{P}_2}(x_{ikl}+x_{ilk})+\sum_{i\in\mathcal{P}_1}\sum_{j\in\mathcal{P}_1}(x_{ijk}+x_{ikj})\\
&+\sum_{i\in\mathcal{P}_1}\sum_{l\in\mathcal{P}_2}(x_{kil}+x_{lik})+\sum_{l\in\mathcal{P}_2}(x_{kl}+x_{lk})+\sum_{i\in\mathcal{P}_1}(x_{ik}+x_{ki})+\sum_{i\in\mathcal{P}_3} x_{ik}+x_k=1, \forall k\in\mathcal{P}_2;
\end{aligned}
\tag{4.3}
$$

$$\sum_{k\in\mathcal{P}_2}\sum_{l\in\mathcal{P}_2} x_{ikl} + \sum_{k\in\mathcal{P}_2\cup\mathcal{P}_4} x_{ik} + x_i = 1, \forall i \in \mathcal{P}_3; \tag{4.4}$$

$$\sum_{i\in\mathcal{P}_1}\sum_{j\in\mathcal{P}_1} x_{ijk} + \sum_{i\in\mathcal{P}_1\cup\mathcal{P}_3} x_{ik} + x_k = 1, \forall k \in \mathcal{P}_4; \tag{4.5}$$

$$\begin{aligned} &V + (W_i + W_j) \leq V_2^m + M[1 - \sum_{k\in\mathcal{P}_2}\sum_{l\in\mathcal{P}_2}(x_{ijkl} + x_{jikl} + x_{ikjl} + x_{jkil}) - \sum_{k\in\mathcal{P}_2\cup P_4}(x_{ijk} + x_{jik}) \\ &- \sum_{k\in\mathcal{P}_2}(x_{ikj} + x_{jki}) - (x_{ij} + x_{ji})], \forall i<j \in \mathcal{P}_1; \end{aligned} \tag{4.6}$$

$$\begin{aligned} &V + (W_i + W_k) \leq V_2^m + M[1 - (\sum_{j\in\mathcal{P}_1}\sum_{l\in\mathcal{P}_2} x_{jkil} - \sum_{j\in\mathcal{P}_1} x_{jki} - \sum_{l\in\mathcal{P}_2} x_{kil} - x_{ki})], \\ &\forall i \in \mathcal{P}_1, \forall k \in \mathcal{P}_2; \end{aligned} \tag{4.7}$$

$$\begin{aligned} &V + (W_k + W_l) \leq V_2^m + M[1 - \sum_{i\in\mathcal{P}_1}\sum_{j\in\mathcal{P}_1}(x_{ijkl} + x_{ijlk} + x_{ikjl} + x_{iljk}) - \\ &\sum_{i\in\mathcal{P}_1\cup P_3}(x_{ikl} + x_{ilk}) - \sum_{i\in\mathcal{P}_1}(x_{kil} + x_{lik}) - (x_{kl} + x_{lk})], \forall k<l \in \mathcal{P}_2; \end{aligned} \tag{4.8}$$

$$\begin{aligned} &v_i \geq A_i + h + t_{o,i} - M[1 - \sum_{j\in\mathcal{P}_1}\sum_{k\in\mathcal{P}_2}\sum_{l\in\mathcal{P}_2}(x_{ijkl} + x_{ikjl}) - \sum_{j\in\mathcal{P}_1}\sum_{k\in\mathcal{P}_2\cup P_4} x_{ijk} - \\ &\sum_{k\in\mathcal{P}_2}\sum_{l\in\mathcal{P}_2} x_{ikl} - \sum_{j\in\mathcal{P}_1}\sum_{k\in\mathcal{P}_2} x_{ikj} - \sum_{j\in\mathcal{P}_1\cup P_2\cup P_4} x_{ij} - x_i], \forall i \in \mathcal{P}_1; \end{aligned} \tag{4.9}$$

$$v_i \geq A_i + h + t_{o,i}, \forall i \in \mathcal{P}_3; \tag{4.10}$$

$$v_k \geq A_i + h + t_{0,k} - M[1 - \sum_{l\in\mathcal{P}_2} x_{kil} - x_{ki}], \forall i \in \mathcal{P}_1, \forall k \in \mathcal{P}_2; \tag{4.11}$$

$$\begin{aligned} &v_i \geq A_j + h + t_{o,i} - M[1 - \sum_{k\in\mathcal{P}_2}\sum_{l\in\mathcal{P}_2}(x_{ijkl} + x_{ikjl}) - \sum_{k\in\mathcal{P}_2\cup P_4} x_{ijk} - \\ &\sum_{k\in\mathcal{P}_2} x_{ikj} - x_{ij}], \forall i \neq j \in \mathcal{P}_1; \end{aligned} \tag{4.12}$$

$$v_j \geq v_i + O + t_{i,j} - M[1 - \sum_{k\in\mathcal{P}_2}\sum_{l\in\mathcal{P}_2} x_{ijkl} - \sum_{k\in\mathcal{P}_2\cup P_4} x_{ijk} - x_{ij}], \forall i \neq j \in \mathcal{P}_1; \tag{4.13}$$

$$v_k \geq v_j + O + t_{j,k} - M[1 - \sum_{i\in\mathcal{P}_1}\sum_{l\in\mathcal{P}_2}(x_{ijkl} + x_{jkil} + x_{iljk}) - \sum_{i\in\mathcal{P}_1}(x_{ijk} + x_{jki}) - \sum_{l\in\mathcal{P}_2}(x_{ljk} + x_{jkl}) - x_{jk}], \forall j \in \mathcal{P}_1 \cup P_3, k \in \mathcal{P}_2 \cup P_4; \tag{4.14}$$

$$v_l \geq v_k + O + t_{k,l} - M[1 - \sum_{i\in\mathcal{P}_1}\sum_{j\in\mathcal{P}_1} x_{ijkl} - \sum_{j\in\mathcal{P}_1} x_{jkl} - x_{kl}], \forall k \neq l \in \mathcal{P}_2; \tag{4.15}$$

$$v_j \geq v_k + O + t_{k,j} - M[1 - \sum_{i\in\mathcal{P}_1}\sum_{l\in\mathcal{P}_2} x_{ikjl} - \sum_{l\in\mathcal{P}_2}(x_{ikj} + x_{kjl}) - x_{kj}], \forall k \in \mathcal{P}_2, j \in \mathcal{P}_1; \tag{4.16}$$

$$T_i^s \leq v_i \leq T_i^e, \forall i \in \mathcal{P}_1 \cup P_2 \cup P_3 \cup P_4; \tag{4.17}$$

$$t(o,i,j,k,l,o)x_{ijkl} \leq T_{max} + M(y_{ijkl}), \forall i,j,k,l; \tag{4.18}$$

$$t(o,i,j,k,o)x_{ijk} \leq T_{max} + M(y_{ijk}), \forall i,j,k; \tag{4.19}$$

$$t(o,i,j,o)x_{ij} \leq T_{max} + M(y_{ij}), \forall i,j; \tag{4.20}$$

$$t(o,i,o)x_i \leq T_{max} + M(y_i), \forall i \in \mathcal{P}_1 \cup P_2 \cup P_3 \cup P_4; \tag{4.21}$$

$$\sum_{i\in\mathcal{P}_1}\sum_{k\in\mathcal{P}_2} x_{ik} + \sum_{i\in\mathcal{P}_1} x_i + \sum_{k\in\mathcal{P}_2} x_k \leq H_1; \tag{4.22}$$

$$\sum_{i\in\mathcal{P}_1}\sum_{j\in\mathcal{P}_1}\sum_{k\in\mathcal{P}_2}\sum_{l\in\mathcal{P}_2}(x_{ijkl} + x_{ikjl}) + \sum_{i\in\mathcal{P}_1}\sum_{j\in\mathcal{P}_1}\sum_{k\in\mathcal{P}_2\cup\mathcal{P}_4} x_{ijk} + \sum_{i\in\mathcal{P}_1\cup\mathcal{P}_3}\sum_{k\in\mathcal{P}_2}\sum_{l\in\mathcal{P}_2} x_{ikl} + \sum_{i\in\mathcal{P}_1}\sum_{k\in\mathcal{P}_2}\sum_{j\in\mathcal{P}_1} x_{ikj} + \sum_{k\in\mathcal{P}_2}\sum_{i\in\mathcal{P}_1}\sum_{l\in\mathcal{P}_2} x_{kil} + \sum_{i\in\mathcal{P}_1}\sum_{k\in\mathcal{P}_4} x_{ik} + \sum_{i\in\mathcal{P}_3}\sum_{k\in\mathcal{P}_2\cup P_4} x_{ik} + \sum_{i\in\mathcal{P}_1}\sum_{j\in\mathcal{P}_1} x_{ij} + \sum_{k\in\mathcal{P}_2}\sum_{i\in\mathcal{P}_1} x_{ki} + \sum_{k\in\mathcal{P}_2}\sum_{l\in\mathcal{P}_2} x_{kl} + \sum_{i\in\mathcal{P}_3\cup\mathcal{P}_4} x_i \leq H_2; \tag{4.23}$$

$$v_i \geq 0, \forall i \in \mathcal{N}; \tag{4.24}$$

$$x_{ijkl}, x_{ijk}, x_{ij}, x_i, y_{ijkl}, y_{ijk}, y_{ij}, y_i \in \{0,1\}, \forall i,j,k,l \in \mathcal{N}; \tag{4.25}$$

The objective (4.1) is to minimize the total travelling cost between the port and customer locations and the total cost of penalty related to the extra driving hours of all trips. Constraints (4.2)-(4.5) forces all orders to be delivered combined or individually. Constraints

(4.6)-(4.8) are designed in order to ensure that the maximum allowance weight is not violated for the total weight of the combined route, which includes the weight of the vehicle, chassis, cargo and containers. The arrival times of combined and individual orders to the final destination is calculated by constraints (4.9)-(4.16). Constraint (4.17) guarantees that the arrival time of the order at its destination is between the assigned earliest and latest time window for each order. Constraints (4.18)-(4.21) ensure that if the maximum allowance working time for a truck route is violated, a penalty is paid in the objective. Constraints (4.22) and (4.23) restrict the total number of 20ft and 40ft trucks used to be less than or equal to the fleet size of each type. Finally, constraints (4.24) and (4.25) determine the domain of variables.

4.3 Heuristic decomposition and aggregation approach

The container combination problem for inland transportation is classified as NP-hard [79], which means exact optimization approaches can only solve limited size instances. In section 4.2 we have formulated a MILP model which solves problems with up to 100 orders in acceptable time by CPLEX. In practice however, a port may operate thousands of containers per day by road. Therefore, an efficient solution approach, for example, heuristic method, is needed. A well designed heuristic has the capability of finding near optimal solutions for large problems in a reasonable time, which in this application is, several hours. In this section a decomposition and aggregation heuristic is developed. The method consists of two stages: firstly decompose orders into small subgroups and find a solution for each subgroup by using the formulated MILP model (See section 4.4 for more detailed results), secondly remove the best (optimally) combined orders and aggregate the remaining orders to form the second stage MILP model. Column generation is applied in both stages to balance fleet sizes assigned to subgroups.

4.3.1 Decomposition of orders

Stage 1 aims to reduce the solution time so as to increase the size of the problem that can be managed. A simple decomposition method based on customer locations is therefore proposed. Considering the aim of combining import and export orders to form a closed delivery route starting and finishing at the port, in this study we deploy the fan-shaped clustering approach [39] to enable geographical decomposition of customer locations. This approach gives the highest possibility of creating closed routes starting and ending at the same depot [103] and therefore is believed to be more suitable than other decomposition strategies such as rectangle-shaped [21], [22], ring-shaped [29], seed-based [31] decompositions etc. To create the decomposition, the polar angle of each order's location with respect to the port and the baseline of the seashore is calculated. Acording to the angles, orders are decomposed into fan-shaped subgroups and MILP like (4.1) is constructed for each subgroup to find the optimal combination within it.

Nevertheless, the decomposition stops the combination of orders between different subgroups, and introduces a question on how to allocate the fleet to serve each subgroup so as to minimize the total delivery cost. It is not hard to imagine, too small fleet sizes may lead to infeasibility and too large fleet may lead to capacity loss. Especially in case when capacity is very limited, how to allocate the fleet amongst service areas forms a vital decision.

4.3.2 Column Generation

In order to balance the fleet size for all subgroups and to decrease the cost gap between the heuristic decomposition and the MILP model, a column generation method is applied. Column generation is an iterative method [32] which consists of two parts: a set of subproblems to generate all columns (potential parts of optimal solution), and a master problem to maintain all generated columns. They are solved iteratively, with the master problem to inform the shadow-price of global constraints and subproblems to identify whether the master problem should be enlarged with additional columns or not. Column generation procedure alternates between the master problem and the subproblem, until the former contains all the necessary optimal columns.

4.3.2.1 Subproblem

Decomposition prevents orders in one subgroup to be combined with the ones in another. Therefore, a separate subproblem can be formulated for every single subgroup. Suppose a subgroup is formed, whether an order is delivered, the arrival time of the order, if it violates the driving regulation and the weight limit are all irrelevant to other subgroups. It follows that the subproblems are formed by removing the ”global constraints” on fleet sizes using Lagrangian Relaxation:

$$
\begin{aligned}
&\min f(x) + f(y) + \pi_1(H_1 - \sum_{i\in\mathcal{P}_1}\sum_{k\in\mathcal{P}_2} x_{ik} + \sum_{i\in\mathcal{P}_1} x_i + \sum_{k\in\mathcal{P}_2} x_k) \\
&\quad + \pi_2(H_2 - \sum_{i\in\mathcal{P}_1}\sum_{j\in\mathcal{P}_1}\sum_{k\in\mathcal{P}_2}\sum_{l\in\mathcal{P}_2}(x_{ijkl} + x_{ikjl}) + \sum_{i\in\mathcal{P}_1}\sum_{j\in\mathcal{P}_1}\sum_{k\in\mathcal{P}_2\cup\mathcal{P}_4} x_{ijk} + \sum_{i\in\mathcal{P}_1\cup\mathcal{P}_3}\sum_{k\in\mathcal{P}_2}\sum_{l\in\mathcal{P}_2} x_{ikl} + \\
&\quad \sum_{i\in\mathcal{P}_1}\sum_{k\in\mathcal{P}_2}\sum_{j\in\mathcal{P}_1} x_{ikj} + \sum_{k\in\mathcal{P}_2}\sum_{i\in\mathcal{P}_1}\sum_{l\in\mathcal{P}_2} x_{kil} + \sum_{i\in\mathcal{P}_1}\sum_{k\in\mathcal{P}_4} x_{ik} + \sum_{i\in\mathcal{P}_3}\sum_{k\in\mathcal{P}_2\cup P_4} x_{ik} + \sum_{i\in\mathcal{P}_1}\sum_{j\in\mathcal{P}_1} x_{ij} + \\
&\quad \sum_{k\in\mathcal{P}_2}\sum_{i\in\mathcal{P}_1} x_{ki} + \sum_{k\in\mathcal{P}_2}\sum_{l\in\mathcal{P}_2} x_{kl} + \sum_{i\in\mathcal{P}_3\cup\mathcal{P}_4} x_i); \\
&s.t. \quad (4.2) - (4.21), (4.24) - (4.25)
\end{aligned}
\tag{4.26}
$$

Where $f(x)$ and $f(y)$ represents the x and y relevant parts in the original objective (4.1), respectively. π_1, π_2 are the Lagrangian Multipliers for constraints (4.22) and (4.23), respectively. Obtaining the optimal solution of (4.1) for every subgroup f_s^*, $s \in \mathcal{A}$, we can generate columns based on whether this new column improves on the existing ones in the RMP. If the solution of all subproblems has non negative reduced cost the iteration is terminated, otherwise the RMP is enlarged by adding more columns and resolved.

4.3.2.2 Restricted Master Problem (RMP)

In this problem, all constraints following decomposition fall in a single subproblem, apart from the fleet size restrictions. We treat them as global constraints which have to be dealt with in the Restricted Master Problem (RMP). In the RMP, columns represent the number of 20ft and 40ft trucks needed by the optimal solution for each subgroup. Each iteration comprises solving the RMP to determine the optimal solution and the dual multipliers.

The RMP can be formulated as follows:

$$\min \sum_{s\in\mathcal{A}} f_s^* \lambda_s \tag{4.27}$$

$$s.t. \quad \sum_{s\in\mathcal{A}} q_s^1 \lambda_s \leq H_1; \tag{4.28}$$

$$\sum_{s\in\mathcal{A}} q_s^2 \lambda_s \leq H_2; \tag{4.29}$$

$$\sum_{s\in\mathcal{A}_k} \lambda_s = 1, \forall k; \tag{4.30}$$

$$0 \leq \lambda_s \leq 1, \forall s \in \mathcal{A}; \tag{4.31}$$

The column set is indicated by $\mathcal{A} = \{\mathcal{A}_k, k = 1, ..., m\}$, which compromises the set of columns generated by subproblem k: $\mathcal{A}_k$. Each column is associated with a variable λ_s, which is restricted between 0 and 1 and indicates the number of times a column is chosen in the solution. q_s^1 and q_s^2 represent the number of 20ft and 40ft trucks required by the optimal solution of solving the corresponding subproblem $s \in \mathcal{A}$.

4.3.3 An example

In this section we demonstrate an example to illustrate how the decomposition-column generation approach works. Geographical locations of the port and a set of 20 orders are shown in Figure 4.4. Assume that the order set comprises : 8×20ft import orders, 6×20ft export orders, 3×40ft import orders and 3×40ft export orders. The available fleet consists of 8×20ft trucks and 6×40ft trucks. The example is firstly solved as a whole by the MILP model, as shown in Figure 4.5 which illustrates the optimal solution.

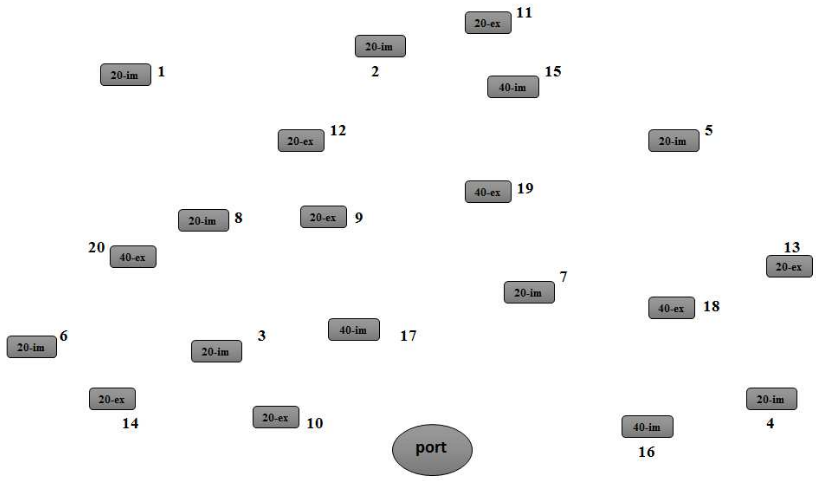

Figure 4.4: *Geographical location of the port and a set of orders*

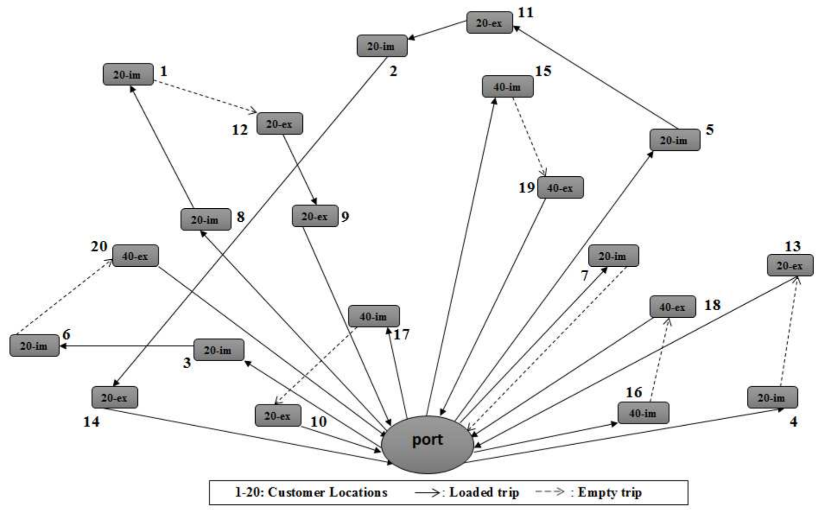

Figure 4.5: *Optimal solution of the example*

In this case, most orders are delivered jointly with others, creating two routes combining four orders, one route combining three orders and four routes combining two orders. Only one order is delivered individually (the order 7). As in Table 4.1, the result shows that the minimum cost is 1194 and the number of the used trucks are 2×20ft and 6×40ft, respectively. We can see when there is no decomposition, the model tends to use as many 40ft trucks as possible to reduce empty trips.

Now we decompose the example into four subgroups like what is shown in Figure 4.6, and solve for each area separately. In this experiment the fleet is simply allocated according to the proportional number of the corresponding container sizes, i.e.

$$\text{no. of 20ft trucks for group } k = [\frac{\text{no. of 20ft orders in group } k}{\text{no. of all 20ft orders}} \times H_1] \tag{4.32}$$

Orders are combined separately within each subgroup and the minimum cost obtained

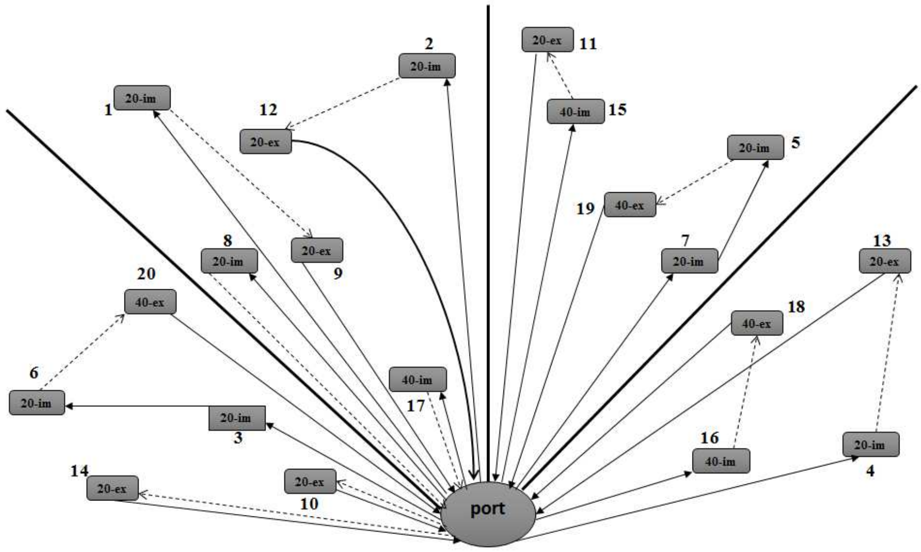

Figure 4.6: *Fan-shaped decomposition and the solution*

from the decomposition solution is 1505, which is 26% higher than the cost of the optimal solution. This is due to the fact that the fleet is not allocated wisely by just considering the proportional number of orders. To automatically balance the fleet sizes amongst subgroups, the column generation approach as stated in section 4.3.2 is applied and the result is shown in Figure 4.7. Note that, decomposing orders as subgroups prevents some orders to be combined with others, so that links between orders 5, 11, 2 and 14 are broken as they fall into three different subgroups. Since trucks are allocated to each subgroup, the four combined routes 8, 1, 12 and 9 are also prevented due to the existence of order 17 in subgroup 2 which has only one 40ft truck in service.

Comparing to the MILP solution, the decomposition uses relatively more 20ft (6×20ft) trucks and relatively less 40ft (5×40ft), although there are 6×40ft available. In this case, one more 40ft truck is allocated to the second subgroup, which enables the optimal route combination of orders 5, 11, 2 and 14 again. This reduces the total case by 11.43% to 1333, and used the full truck load of four out of six 40ft trucks.

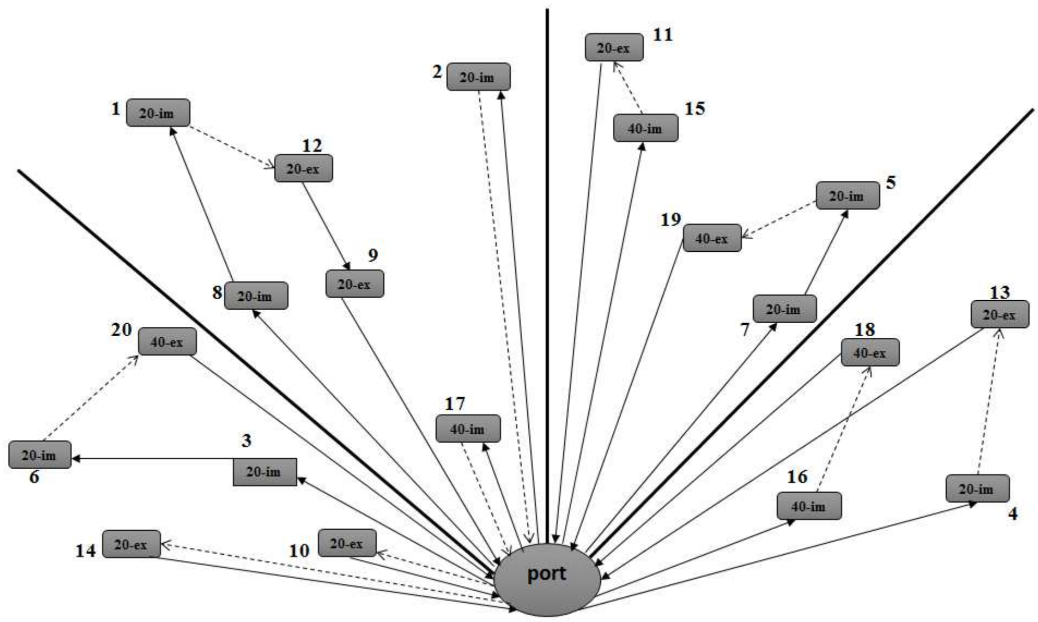

Figure 4.7: *Fan-shaped decomposition with column generation and the solution*

It is clear from the result that the cost is still higher than the MILP, therefore we will develop an aggregation step to reduce the problem size and the cost as will be explained later.

# Orders	group index	# Orders in group	# fleet available 20ft	# fleet available 40ft	# fleet used 20ft	# fleet used 40ft	# Indiv. routes	# 2 combined routes	# 3 comb. routes	# 4 comb. routes	Min. cost of groups	Total cost	CPU Time(sec)
MILP - Optimal solution													
20	-	-	8	6	2	6	1	4	1	2	-	1194	03:22
decomposition solution													
	1	5	2	1	2	1	2	-	1	-	241		
	2	6	3	1	3	1	2	2	-	-	613		
20	3	5	2	2	-	2	-	1	1	-	508	1505	01:76
	4	4	1	2	1	1	-	2	-	-	143		
decomposition with column generation solution													
	1	5			2	1	2	-	1	-	241		
	2	6			1	2	2	-	-	1	441		
20	3	5	8	6	-	2	-	1	1	-	508	1333	03:44
	4	4			1	1	-	2	-	-	143		

Table 4.1: *Results for the optimal solution, decomposition and decomposition with column generation of the example*

4.3.4 Aggregation of orders

Looking at the previous example we can see that the decomposition does introduce significant gaps to the real optimal solution, since it prevents the combination of orders across subgroups. In this section we will discuss how to aggregate to create more combined deliveries so as to further reduce cost. Obviously, in the decomposed optimal solution some routes are already dense and have fully used the truck load, e.g. the routes combining four 20ft orders. For this type of route, relaxing the boundary of decomposition may shift part of the route combination to its neighbour area but would not significantly reduce the delivery cost since orders in the same subgroup are believed to be closer to each other. Nevertheless, orders that are loosely combined with others or equivalently, orders that are delivered on routes that still have spare capacity for accommodating one/several more orders, should be the ones to be considered further. Therefore, in the second stage, we remove the "well-combined" orders from the list, which include:

- 4×20ft
- 1×40ft + 2×20ft

- 2×40ft

and aggregate all the remaining orders to solve another MILP. By doing this, the size of the problem is largely reduced and so is the computation time. Based on the above explanation of the aggregation, orders 1,3, 5, 6, 7, 8, 9, 12, 16, 18, 19 and 20 are removed from the optimal solution of the decomposed model for the previous example where the cost to deliver them is 651 (see Table 4.2 for more information). This leaves 8 orders for the aggregation problem. Here we consider 2 options for aggregation:

- Aggregate every two adjacent subgroups as in Figure 4.8.
- Aggregation as a whole as in Figure 4.9.

Note that these are the two extreme ways of doing aggregation, which can also be done at other levels in between. According to the real data, the best aggregation strategy can be selected by experience considering customer density, driving time, travelling speed and solution time for the MILP. The result in Table 4.2 shows that the total minimum cost of the aggregation as two subgroups is 1245, which is the same as aggregating as a whole. It can be seen that in both cases the cost is about 7% smaller than the decomposition with column generation, and being very close to the real optimal solution (just 4% higher).

Comparing the optimal delivery routes of the MILP and that of the decomposition-and-aggregation method, many routes are the same except the routes relating to orders 2, 5, 7, 11, 14, 15 and 19. In the optimal solution these orders form a 4-combined route of 20ft orders, a 2-combined route of 40-ft orders and a single delivery route, whereas in the heuristic solution they form a 3-combined and two 2-combined routes. The used fleet size

# Orders	# groups	# of groups	# fleet available 20ft	40ft	used 20ft	40ft	# Indiv. routes	# 2 combined routes	# 3 comb. routes	# 4 comb. routes	Min. cost	Cost of Removed orders	Total cost	CPU time(sec.)
decomposition with column generation solution														
	1	5			2	1	2	-	1	-	241			
	2	6			1	2	2	-	-	1	441			
20	3	5	8	6	-	2	-	1	1	-	508	-	1333	02:44
	4	4			1	1	-	2	-	-	143			
aggregation of every two adjacent subgroups solution														
8	1	4			1	1	-	2	-	-	218			
	2	4	8	2	1	1	-	2	-	-	376	651	1245	03:31
aggregation as a whole solution														
8	1	8	8	2	2	2	-	4	-	-	594	651	1245	04:65

Table 4.2: *Results for the aggregation of the example*

varies in different methods. However, in all methods, all available 40ft trucks tend to be used rather than the 20ft trucks to reduce the total cost.

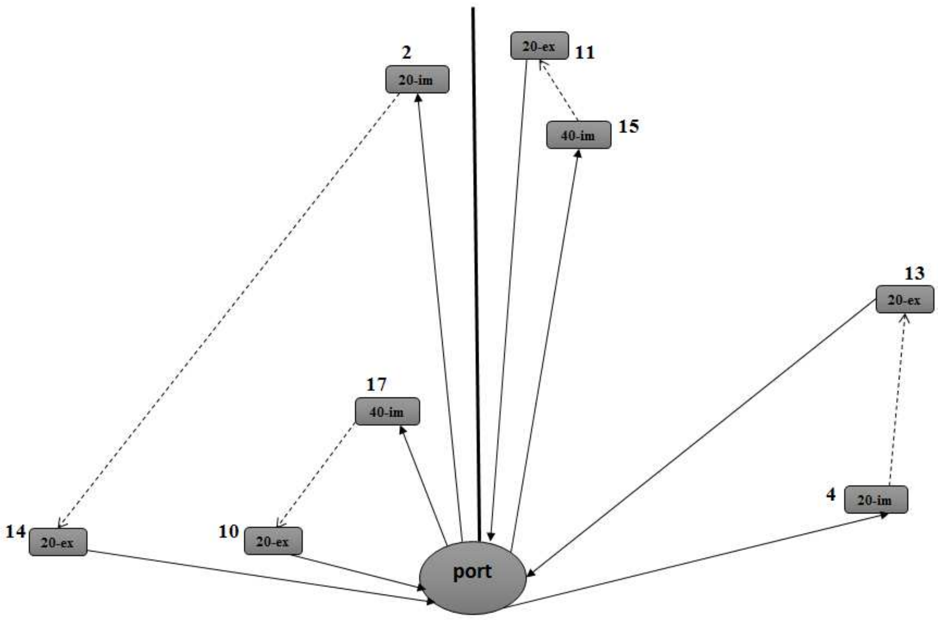

Figure 4.8: *Aggregation of orders as two subgroups*

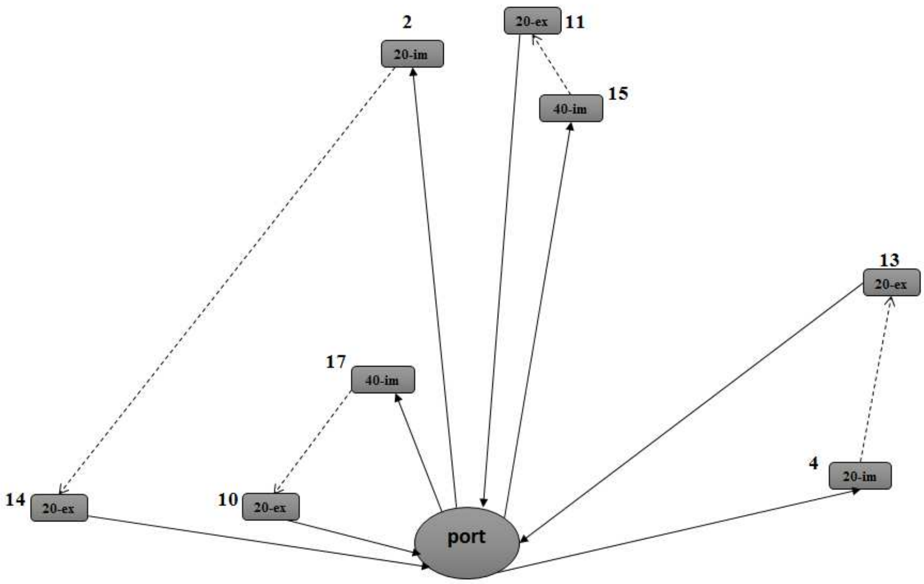

Figure 4.9: *Aggregation of orders as a whole*

4.4 Numerical Results

In this section we present the numerical results for the mixed integer linear programming (MILP) model, the heuristic decomposition-column generation and the decomposition-aggregation approach. Note that both methods are coded in MATLAB R2015b and executed via CPLEX 12.6.1, on a CPU with an Intel(R)Core(TM)i7-4790 processor.

4.4.1 Results for the MILP model

Geographical data on customer locations are simulated from the real service area of the Port of Felixstowe, which is the largest container port in the UK. A haulage company, who holds a number of 20ft and 40ft trucks is assumed to fulfil the import and export container orders starting and finishing at the port terminal. Distances between the port and customers' locations and between each pair of customers are calculated based on the straight-line distance and rescaled to approximate the road distance. According to the UK

#	Number and Type of Containers				# fleet				#	# 2	# 3	# 4	Min.	CPU
Orders	# 20ft Import	# 20ft Export	# 40ft Import	# 40ft Export	available		used		Indiv. routes	combined routes	combined routes	combined routes	cost	time(sec.)
					20ft	40ft	20ft	40ft						
10	3	3	2	2	3	2	3	2	-	5	-	-	795	00:01
	4	2	1	3	2	3	2	3	2	1	2	-	559	00:15
	5	3	1	1	4	2	3	2	2	1	2	-	507	00:09
20	8	6	3	3	6	4	6	4	2	6	2	-	1227	01:69
	7	8	2	3	5	5	4	5	2	3	4	-	1175	02:72
	9	7	2	2	7	3	7	3	2	6	2	-	1228	04:13
50	15	15	10	10	15	16	3	16	-	8	10	1	2521	1183:69
	17	13	9	11	18	17	1	17	-	8	6	4	2458	1081:71
	14	16	11	9	16	16	4	16	1	9	9	1	2526	1099:93
80	20	20	20	20	25	20	20	20	-	40	-	-	5871	11491:28
	22	18	22	18	20	25	16	25	8	28	4	1	4672	11171:65
	25	15	18	22	23	23	15	23	2	30	6	-	4549	9234:34
100	25	25	25	25	30	30	15	30	-	36	8	1	7220	75443:75
	20	30	20	30	35	35	20	35	19	28	7	1	8370	63247:46
	23	23	26	28	33	38	3	38	2	24	10	5	7454	48186:13
120	40	40	20	20	40	35	-	-	-	-	-	-	-	Out
	45	35	15	25	42	30	-	-	-	-	-	-	-	of
	35	45	25	15	35	40	-	-	-	-	-	-	-	Memory

Table 4.3: *Results for testing the MILP model on different scale of instances.*

working driving regulation, a penalty cost (200 pounds/h) applies for any extra working hours driven. Average speed for trucks is estimated at (40) mile/h. It is assumed that the available time for the collection of import containers is known in advance and the service time at customers' locations is approximated to 2 hours. A variant number of orders of 20ft and 40ft import/export containers and different fleet sizes represented by 20ft and 40ft trucks under service are tested.

As in Table 4.3, the MILP is tested for different size of instances with 10, 20, 50, 80, 100 and 120 orders. For each instance, different types of orders and fleet sizes are applied. The result shows that the model constructs some routes by combining four orders (2×20ft import orders + 2×20ft export orders) in some instances, especially when the number of orders increases and also the density. It is also clear from the result that the model always intends to use the full available 40ft fleet rather than the 20ft trucks, due to the fact that

# Orders	Number and Type of Containers				# fleet				# Indiv. routes	# 2 combined routes	# 3 combined routes	# 4 combined routes	Min. cost	Penalty cost	CPU time(sec.)
	# 20ft Import	# 20ft Export	# 40ft Import	# 40ft Export	available		used								
					20ft	40ft	20ft	40ft							
Results for smaller fleet size															
	15	15	10	10	15	10	15	10	-	25	-	-	3430	-	958:87
50	17	13	9	11	12	12	11	12	-	20	2	1	3050	-	1073:99
	14	16	11	9	10	15	6	15	1	12	7	1	2641	-	1116:46
Results for increased distances															
	15	15	10	10	15	16	3	16	-	9	8	2	6124	400	1150:10
50	17	13	9	11	15	16	3	16	-	9	8	2	6158	400	1140:32
	14	16	11	9	16	16	4	16	1	9	9	1	7229	800	1129:50

Table 4.4: *Results for smaller fleet size and increased distances.*

doing pairwise delivery on 40ft trucks reduces the number of empty trips so as to reduce the total delivery cost. It is obvious from the result that the computation time increases dramatically with the size of the numerical instances. Within each instance category, the smaller the number of 20ft orders under consideration the quicker to find the optimal solution. Nevertheless, the MILP solves to optimality instances with up to 100 orders in 21 hours, which is already larger than what can be managed using the VRP-SPDTW model which normally solves instances with up to 63 orders [97].

In order to test the sensitivity of the MILP, the model is applied to two modified cases on 50 container examples. In the first case, a smaller fleet size is used and as seen in Table 4.4, less 4 combined routes are constructed which leads to a larger delivery cost compared to the original case in Table 4.3. While in the second case, distances between the port and customers' locations are increased. In this case as shown in Table 4.4, the total cost is larger since a penalty cost should be paid for extra working time particulary for the 4 combined routes which take a longer time to complete.

4.4.2 Results for heuristic decomposition and aggregation approach

In this section, the results of the decomposition and aggregation method will be explained. To find the weakness from the optimal solution obtained with the MILP model, the same data of the two instance scales: 50 and 100 orders are tested, each has three different examples. Results of applying the decomposition (D), the decomposition-column generation (D+CG) and the decomposition-and-aggregation (D+CG+Agg) are summarised in Table 4.5. In all test examples, orders are allocated into five subgroups in the decomposition stage, and assembled into one in the aggregation stage. The number of orders in every subgroup varies, but due to the fact that the decomposition is done according to the polar angle, the proposed scheme works better than dividing orders evenly into subgroups.

Note that in the decomposition step (D), we use the same proportional heuristic to split the fleet into subgroups, as stated in formula (4.32). The column generation step (D+CG) actively reassign fleets, especially the 40ft trucks, amongst subgroups to construct more cost-saving routes. For example, in Instance 50-1, increasing the number of 40ft trucks from 6 to 7 in subgroup 4 reduces the subgroup cost by 181, meanwhile taking the required 40ft truck from subgroup 5 increases its cost by 150. Therefore 31 is saved in total by this fleet transition. In another example, Instance 100-1, decomposition with the proportional heuristic for fleet allocation (D) creates an infeasible solution for subgroup 1 due to the assignment of inadequate 20ft trucks, although spare ones are actually available in other subgroups. Implementing the column generation step this lack of trucks is quickly sorted out, which creates at least a feasible delivery plan.

Looking at the result we can see, doing only decomposition, even with column generation to balance fleets, increases the total cost by 15.1-28.7% on 50 orders instances and

Instance	group	# Orders in group	# fleet				# Indiv.	# 2 combined	# 3 comb.	# 4 comb.	Min.	Cost of	Total	(%) worse	CPU
			available		used										
	index		20ft	40ft	20ft	40ft	routes	routes	routes	routes	cost	Removed orders	cost	than MILP	time(sec.)
50-1															
MILP	-	-	15	16	3	16	-	8	10	1	2521	-	2521	-	1183:69
	1	2	1	-	1	-	-	1	-	-	89				
	2	5	2	2	1	2	2	-	1	-	264				
D	3	5	1	3	-	3	1	2	-	-	273	-	3275	29.9	03:03
	4	24	8	6	8	6	5	8	1	-	1841				
	5	14	3	5	1	5	1	3	1	1	808				
	1	2			1	-	-	1	-	-	89				
	2	5			1	2	2	-	1	-	264				
D+CG	3	5	15	16	-	3	1	2	-	-	273	-	3244	28.7	20:88
	4	24			6	7	6	3	4	-	1660				
	5	14			3	4	2	4	-	1	958				
D+CG+Ag	1	19	15	4	4	4	1	4	2	1	816	1781	2597	3.0	75:44
50-2															
MILP	-	-	18	17	1	17	-	8	6	4	2458	-	2458	-	1081:71
	1	27	10	9	4	9	4	5	3	1	1737				
	2	6	2	3	-	3	-	3	-	-	233				
D	3	1	1	-	1	-	1	-	-	-	59	-	3029	23.2	05:03
	4	1	1	-	1	-	1	-	-	-	39				
	5	15	4	5	2	5	2	3	1	1	961				
	1	27			2	10	3	4	4	1	1626				
	2	6			2	2	2	2	-	-	295				
D+CG	3	1	18	17	1	-	1	-	-	-	59	-	2980	21.2	20:88
	4	1			1	-	1	-	-	-	39				
	5	15			2	5	2	3	1	1	961				
D+CG+Ag	1	13	18	3	2	3	-	3	1	1	522	2132	2654	8.0	35:34
50-3															
MILP	-	-	16	16	4	16	1	11	5	3	2537	-	2537	-	1116:74
	1	14	3	6	1	6	1	3	1	1	925				
	2	5	2	2	1	2	1	2	-	-	191				
D	3	1	1	-	1	-	1	-	-	-	44	-	3031	19.5	04:03
	4	1	1	-	1	-	1	-	-	-	20				
	5	29	9	8	7	8	4	8	3	-	1851				
	1	14			2	5	1	5	1	-	1026				
	2	5			3	1	3	1	-	-	250				
D+CG	3	1	16	16	1	-	1	-	-	-	44	-	2921	15.1	22:73
	4	1			1	-	1	-	-	-	20				
	5	29			3	10	4	4	3	2	1581				
D+CG+Ag	1	14	16	2	4	2	-	5	-	1	641	2015	2656	4.7	61:16
100-1															
MILP	-	-	30	30	15	30	-	36	8	1	7220	-	7220	-	75443:75
	1	8	3	2	-	-	-	-	-	-	Inf.				
	2	15	4	5	4	5	3	6	-	-	1254				
D	3	31	9	10	5	10	3	9	2	1	2399	-	Inf.	-	10:13
	4	31	9	10	5	10	1	12	2	-	2724				
	5	14	5	3	4	3	1	5	1	-	1132				
	1	9			4	2	3	3	-	-	513				
	2	15			4	5	3	6	-	-	1254				
D+CG	3	31	30	30	5	10	3	9	2	1	2399	-	8022	11.1	57:83
	4	31			5	10	1	12	2	-	2724				
	5	14			4	3	1	5	1	-	1132				
D+CG+Ag	1	39	30	3	15	3	-	15	3	-	2772	4607	7379	2.2	4245:00
100-2															
MILP	-	-	35	35	20	35	19	28	7	1	8370	-	8370	-	63247:46
	1	9	4	3	2	3	2	2	1	-	411				
	2	12	4	4	2	4	1	4	1	-	888				
D	3	23	7	9	3	9	3	7	2	-	1507	-	8789	5.0	11:23
	4	32	11	11	9	11	11	7	1	1	3560				
	5	24	9	8	7	8	8	5	2	-	2423				
	1	9			4	2	3	3	-	-	513				
	2	12			4	3	2	5	-	-	1050				
D+CG	3	23	35	35	3	9	3	7	2	-	1507	-	8626	3.1	54:45
	4	32			7	12	9	7	3	-	3345				
	5	24			5	9	7	4	3	-	2211				
D+CG+Ag	1	44	35	11	20	11	19	11	1	-	4074	4364	8438	0.8	5940:00
100-3															
MILP	-	-	38	33	3	33	2	24	10	5	7454	-	7454	-	48186:13
	1	9	4	2	4	2	3	3	-	-	513				
	2	15	5	6	1	6	1	5	-	1	960				
D	3	31	12	10	4	10	-	11	3	-	2323	-	7768	4.2	13:52
	4	31	10	12	4	12	3	12	-	1	2840				
	5	14	7	3	4	3	1	5	1	-	1132				
	1	9			4	2	3	3	-	-	513				
	2	15			3	5	1	7	-	-	1119				
D+CG	3	31	38	33	4	10	-	11	3	-	2323	-	7726	3.7	51:35
	4	31			4	12	3	12	-	1	2840				
	5	14			2	4	1	2	3	-	931				
D+CG+Ag	1	34	38	4	13	4	3	11	3	-	2350	5156	7506	0.7	4040:00

Table 4.5: *Results of all Methods for 50 and 100 orders*

3.1-11.1% on 100 orders instances. The suboptimality gap is reducing as problem size goes up, since the denser the orders are, the better chance to construct combined routes in each subgroup and therefore the closer the solution is to the optimal one. On the other hand, by doing the aggregation the total cost is reduced to 3.0-8.0% higher than the MILP optimal solution for 50 orders and to only 0.7-2.2% higher for 100 orders examples, which means the resulting solutions are very close to optimality. However, the time it takes to perform the decomposition-and-aggregation approach is much less than solving the MILP. Instances with 50 orders are solved in 1.25 minutes while the MILP needs about 20 minutes, and instances with 100 orders are managed in 1.65 hours which is just 7% of the CPLEX time of 21 hours.

4.4.3 Results for different decomposition levels

In this section we test the different levels of decomposition-and-aggregation approach, to see the tradeoff between the performance and the solution time. Here we take a problem with 200 orders for example. From Table 4.3 we can see the solution time of a 20-order example is around 2.8s and for 50-order ones is around 1121s. When going above 50, the solution time increases very fast to hours. Therefore in this test, we consider the decomposition levels leading to roughly 20 or 50 orders in every subgroup, which means decomposing the entire problem into 9 and 5 subgroups, respectively. Similarly in the aggregation stage, to avoid a very long solution time, we choose the aggregation level that yields up to 50 orders per aggregated group, and solve the resulting MILP to see the influence of different decomposition levels. As seen in Table 4.6, decomposing into more subgroups definitely increases total cost, in all three stages of the solution process. Nev-

Method	# group index	# Orders of groups	# fleet available 20ft	# fleet available 40ft	# fleet used 20ft	# fleet used 40ft	# Indiv. routes	# 2 combined routes	# 3 comb. routes	# 4 comb. routes	Min. cost	Cost of Removed orders	Total cost	CPU time(sec.)
200- 5 groups														
	1	38	12	9	10	9	7	7	3	2	2619			
	2	39	11	13	1	13	-	6	5	3	2669			
D	3	45	12	15	2	15	2	4	9	2	3145	-	15076	4338:00
	4	42	13	10	12	10	9	7	5	1	4277			
	5	36	12	8	9	8	4	9	2	2	2366			
	1	38			10	9	7	7	3	2	2619			
	2	39			1	13	-	6	5	3	2669			
D+CG	3	45	60	55	6	13	2	8	9	-	3423	-	14986	47812:88
	4	42			6	13	6	5	6	2	3754			
	5	36			11	7	4	11	2	1	2521			
	1	22			13	2	8	7	-	-	1555			
D+CG+Ag	2	43	60	8	17	6	8	12	1	2	3166	9765	14486	52150:44
200- 9 groups														
	1	16	6	2	6	2	2	4	2	-	1324			
	2	27	7	9	4	9	4	6	1	2	1894			
	3	24	6	8	1	8	-	4	4	1	1694			
	4	16	5	4	3	4	1	3	3	-	1415			
D	5	30	7	11	-	11	-	5	4	2	2207	-	15957	180:00
	6	25	8	6	8	6	7	5	-	2	2318			
	7	24	8	6	6	6	5	3	3	1	2458			
	8	16	6	3	6	3	5	2	1	1	1313			
	9	22	7	6	4	6	2	5	2	1	1334			
	1	16			6	2	2	4	2	-	1324			
	2	27			4	9	4	6	1	2	1894			
	3	24			1	8	-	4	4	1	1694			
	4	16			1	5	1	1	3	1	1234			
D+CG	5	30	60	55	2	10	1	6	3	2	2207	-	15732	565:00
	6	25			6	7	7	2	2	2	2286			
	7	24			4	7	4	2	4	1	2130			
	8	16			7	3	5	4	1	-	1474			
	9	22			7	4	2	7	2	-	1489			
	1	28			13	4	8	7	2	-	1930			
D+CG+Ag	2	40	60	9	17	5	8	12	-	2	2952	9905	14787	4421:00

Table 4.6: *Results of all Methods for varies number of groups of 200 orders*

ertheless, the final decomposition-and-aggregation result for the 9-group decomposition is just 2% higher than that is for the 5-group, while the latter takes 14.5 hours to achieve compared to 1.2 hours for the former. Therefore in practical applications, decomposing into subgroups with 20-30 orders achieves a good balance between the solution time and optimality.

4.4.4 Results for larger instances

Finally, a set of large size instances: 120, 150, 200, 300 and 400 orders are tested. Table 4.7 summarises the result of solving the large size instances by the three processes. For each instance, orders are decomposed into a different number of subgroups based on the resulting number of orders in each subgroup. For example, the 120 and 150 orders are

# Orders	# groups	Decomposition		Decomp. + Column Generation			Decomp. + Column Gen. + Aggregation		
		time (sec.)	cost	time (sec.)	cost	Saving (%)from Decomposition	time (sec.)	cost	Saving (%) from Decomposition
120	6	23:00	11490	72:00	10855	5.5	2086:00	10026	12.7
150	6	169:00	13751	380:00	12992	5.5	1195:00	12369	10
200	9	180:00	15957	565:00	15732	1.4	4421:00	14787	7.3
300	12	665:00	23876	7588:00	22870	4.2	10919:00	21582	9.6
400	15	788:00	35000	12435:00	33767	3.5	22216:00	30941	11.6

Table 4.7: *Results for larger instances*

decomposed into 6 subgroups, while the 200, 300, 400 are decomposed into 9, 12 and 15 subgroups, respectively. The average cost saving of solving the different instances with the decomposition-column generation is 4.02%, while the average improvement after doing the aggregation is 10.24%. Examples with up to 200 orders can be managed efficiently in 1.2 hours. For very large instances the solution time is still high. It can be seen that the solution time increases linearly with problem size for the Decomposition-only approach. Therefore if the solution time is very restricted, one can also divide the orders together with the fleets completely into parts, and solve each part using the Decomposition-and-aggregation approach independently from others.

4.5 Summary

This chapter investigates the transportation of two types of orders: 20ft and 40ft containers on road with both 20ft and 40ft long trucks. For this purpose a Mixed Integer Programming (MILP) model is proposed. The aim of this model is to minimize the travelling cost and penalty cost for violating the working time regulation. The most important practical restrictions are captured in this model such as the collection time of containers at the port/terminal, the regulation of the working time and the time windows at customers' locations. The MILP model can be implemented to obtain the decision of delivering orders as: 4 combined orders, 3 combined orders, 2 combined orders or individual delivery based on the aforementioned restrictions of the problem. The MILP model is tested for different size of instances drawn from real geographical data in which the result shows that the model is capable to solve efficiently the problem with up to 100 orders by using the CPLEX software package.

In order to deal with larger instances, a decomposition and aggregation heuristic approach is designed in which the locations of orders are decomposed geographically into small subgroups and solved by the formulated MILP model. In order to balance the fleet size amongst all subgroups, a column generation method is used. Orders are then aggregated by removing the best combined orders, in order to reduce the problem size and to improve the cost obtained from the MILP and the initial decomposition problems with up to 400 orders are solved with this methodology. The result shows that the suboptimality of the proposed decomposition-and-aggregation approach is between 0.7-8.0% on the test examples, but consumes just 7% of the CPLEX solution time. For very large instances where the optimal solution is not achievable, results are compared with an industrial-standard simple decomposition approach. The proposed method improves this decomposition cost by 10% on average, which is also easily mergeable with the simple decomposition to meet practical needs on the solution time and accuracy.

Chapter 5

Strip and Discharge of Containers

5.1 Introduction

In inland transportation, customers are classified into two groups, importers who receive loaded containers from the port/terminal and exporters who send the loaded containers to the port. On the other hand, empty containers should be moved between importers/exporters, the port and container depots based on the required supply and demand. In the container industry, some shipping lines own a fleet of trucks to perform the transportation, while some others outsource inland delivery to local haulage companies. This chapter considers the former case where a shipping line can decided where to collect/drop the empty containers that are demanded at some customer locations and how many empty boxes to collect/drop at all inland depots.

In the container industry, an *order* can be defined as a request to deliver a 20ft or 40ft equivalent cargo from its origin to destination, in containers. In practice, two major types of container loading/unloading rules are deployed, say the "*Strip*" and "*Discharge*". In the

Strip case, the container, together with its cargo (if not empty), is put onto/removed from the carrying truck at customer locations. The customer then deals with the container by loading/removing its cargo and make another request for the pick-up/removal of empty/loaded containers. This process will generate many empty requests, and normally the customer does not care about where the empty box comes from/goes to. While in the *Discharge* case, only the cargo will be loaded onto/unloaded from the delivery truck; containers and trucks are not separated after services. Traditional studies separate these two types of delivery mode when planning the route. However, there is a great potential for merging all possible types together to reflect the practical situation, as the handling of one type does not affect the other given there is relevant equipment at the customer location. In this work, we propose a methodology to maintain the joint delivery of both types, so as to reduce the overall transportation cost.

The combination of containers for inland transportation is classified as NP-hard [79], thus even managing one type (*Strip* or *Discharge* of containers) or the joint case is also NP-hard. In this case exact optimization approaches are believed to be time prohibitive and can only solve limited size instances. For this purpose, a Genetic Algorithm (GA) approach for combining the inland transportation of heterogenous (20ft and 40ft) loaded and empty containers is developed in this chapter. We investigate the delivery process of 12 different order types, which covers all types of container transportation modes in practice, between the port/depots and customer locations using a heterogeneous fleet consisting of both 20ft and 40ft trucks. Both common modes of the container transportation, say *Strip* and *Discharge*, are considered in this research. The two cases are investigated separately and jointly to evaluate the benefit of combining these two types of containers delivery in reality. The genetic algorithm approaches are also designed based on the two common strategies for managing the empty containers, say the *Street-turn* where empty containers are delivered directly from import customers to serve exporter customers, and the *Depot-turn* where empty containers can be stored at/collected from inland depots for later reuse. We also evaluate the usage of inland depots as a means of saving cost in empty container management. A comparison of numerical experiments indicates that it is worthwhile to combine all these types of transportation with the existence of the inland depots. In addition, inland empty depots help saving delivery cost by allowing more flexible delivery routes and improving truck utilization. The developed GA method solves large size instances (1000 orders) in about 500 seconds.

The structure of this chapter is as follows. The problem statement is described in Section 5.2. The genetic algorithm is demonstrated in Section 5.3, and the computational results are presented in Section 5.4. Section 5.5 presents summary of this chapter.

5.2 Problem description

In this research, we consider a shipping line managing the transportation of orders (loaded/ empty containers) between the port, inland depots and customers (importer/exporter). Two common types of trucks, i.e. 20ft truck which carries a single 20ft container at a time and 40ft truck which carries a single 40ft container, a single or two 20ft containers at a time are selected to perform the delivery. All routes are supposed to start from and end at the port; the leading cost of travelling from the home depot of the trucks to the port is ignored. We consider both *Strip* and *Discharge* orders, by allowing them to be delivered jointly on the same route/truck if it is profitable.

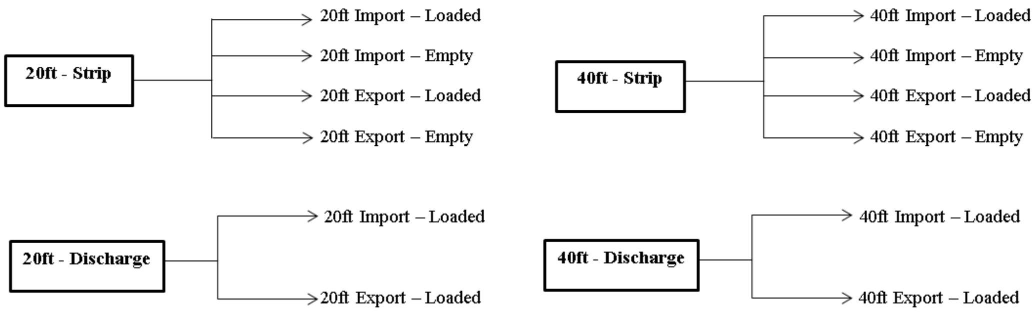

Figure 5.1: *Types of orders*

The full list of order types we are considering is summarised in Figure 5.1. As we can see there are 12 different order types, covering loaded, empty; 20ft, 40ft; *Strip*, *Discharge*, which covers all types of container transportation modes in practice. The developed methodology will allow the combination of all these types to construct the most cost efficient route for this very practical case.

Given the existence of empty containers, there are two major strategies for storing and relocating empty containers: *Depot-turn* and *Street-turn*. In the *Depot-turn* case (see Figure 5.2), empty containers are stored in the port and/or inland depots. All empty container movements have to originate from or terminate at a depot/port, rather than be moved directly from an empty supplier (normally an importer) to an empty demander (normally an exporter).

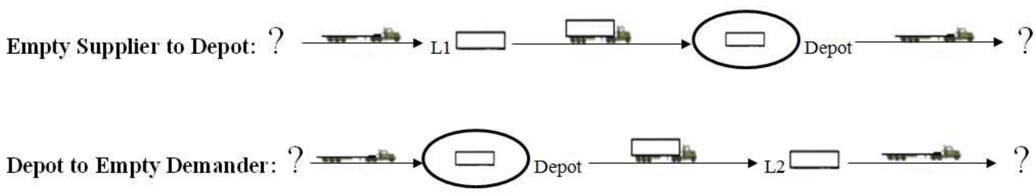

Figure 5.2: *Empty movement with Depot-turn*

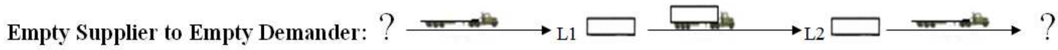

Figure 5.3: *Empty movement with Street-turn*

However, in the *Street-turn* case (see Figure 5.3), the direct movement of empty containers between two customer locations is allowed. It is obvious that the *Depot-turn* is easier to manage, since the closest container depot for every customer location can be identified beforehand, so all of the empty movement requests come with a fixed origin and destination. On the other hand, the *Street-turn* case is more efficient, due to the obviously less number of movements one has to perform. In this research, we allow both types of empty movements (*Street-turn* or *Depot-turn*), as long as the delivery route is feasible and cost saving.

Note that in the *Depot-turn* case, how many empty containers are available at each depot, or how to balance the number of empty containers at depots to meet future needs, are out of the scope of this research. We simply select the least cost depot to drop/collect the empty container (which might be different from the closest depot from/to the customer who is requesting the empty, depending on the actual route the vehicle is travelling), and assume all empty container depots have infinite/sufficient number of storage/capacity available.

To demonstrate some types of orders we are managing in this research, we give some examples in Table 5.1 and elaborate their transportation with figures. For instance, order 1 is an Import-Full (IF) *Strip* order, which is to be picked from the port and delivered to location L1. As shown in Figure 5.4 (Order 1), a truck carrying order 1, which is a loaded container, travels from the port to location L1, at where the container is removed together with its cargo and the truck leaves to somewhere else to serve a new order. The cargo is then removed from the container by the customer which leaves an empty container at L1

Order	Origin	Destination	Type	Payload Weight(kg)	Size	Demonstration
1-IF	Port	L1	Strip	13000	20ft	Figure 5.4
2-IE	L1	NULL	Strip	0	20ft	Figure 5.4
3-IF	Port	L2	Discharge	15000	20ft	Figure 5.5
4-IF	Port	L1	Strip	13500	20ft	Figure 5.6
5-IF	Port	L2	Discharge	15500	20ft	Figure 5.6
6-EE	NULL	L3	Strip	0	20ft	Figure 5.6
7-EF	L4	Port	Discharge	20000	40ft	Figure 5.6

Table 5.1: *Input of orders for the Strip, Discharge, Depot and Street turn cases*

for later collection. To complete the service, Order 2, an Import-Empty (IE) order, is then generated to remove the empty container from L1, which has zero payload weight and no designated destination. It can be relocated to a depot for later usage, or directly to an empty demander to meet another request. In contrast to the *Strip* case, order 3 is a *Discharge* IF order. A truck carrying it from the port to location L2, unloads its cargo at the customer location and then the same truck moves away carrying the emptied container (see Figure 5.5).

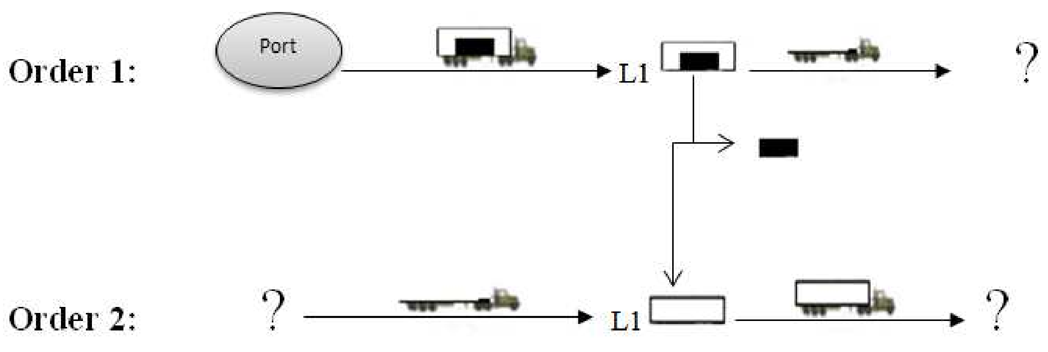

Figure 5.4: *Strip of Orders*

As said earlier, in this work we consider the joint delivery of all types of orders (*Strip* and *Discharge*) following both the *Depot-turn* and the *Street-turn* empty strategies. An example

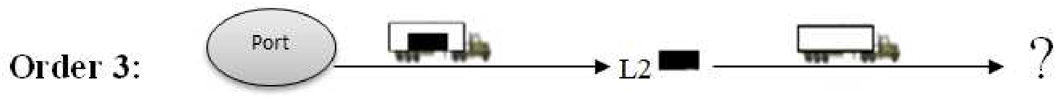

Figure 5.5: *Discharge of Orders*

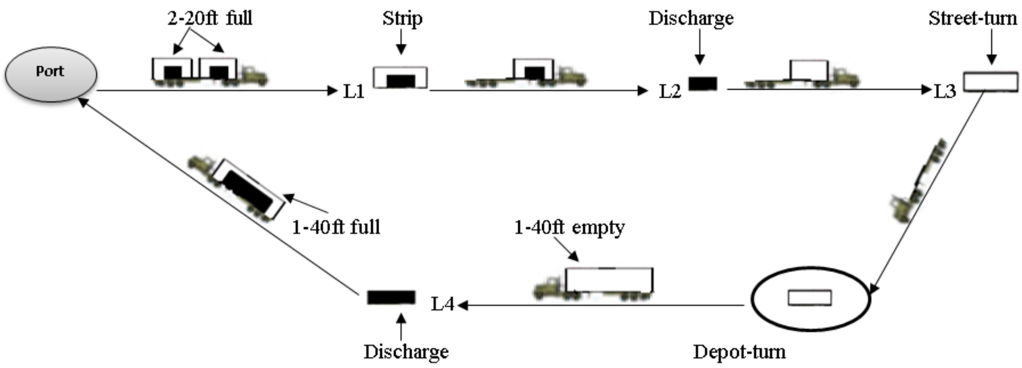

Figure 5.6: *An example of combined delivery route*

route is shown in Figure 5.6, where the truck performs 20ft IF *Strip* (Order 4) and *Discharge* (Order 5) orders to L1 and L2, followed by a *Street-turn* empty (Order 6) movement between L2 and L3, then collect another empty container from an inland depot (*Depot-turn*) to meet an EF *Discharge* demand (Order 7) at L4. We see that all delivery possibilities are allowed to be considered and combined in the same route and by the same truck. Based on the description of the problem, we aim to manage the combination of the different types of *Strip* and *Discharge* orders including the *Street-turn* and *Depot-turn* cases within the allowed daily working time. The objective is to optimize the delivery plan in this general and mixed context to minimize the total transportation cost. Indeed, we aim to minimize the distance travelled by each truck to avoid paying a penalty for violating the allowed working time.

5.3 Genetic algorithm approach

The idea of a genetic algorithm (GA) was first proposed by John Holland [50] as a biological evolution process for life cycle. Later, GA has been used as an optimization technique for solving difficult combinatorial optimization problems [25,47,81]. To solve a problem using GA, an initial generation of solutions are randomly created based on the nature of the problem. These solutions are then evaluated by defining a specific fitness function which usually represents the objective function and results of certain feasibility checks. Then the genetic operators (crossover and mutation) are applied to create offsprings of the selected solutions in the current generation, based on their fitness values. The process continues until the best solution is found (when known) or a certain maximum number of generations have been examined.

The inland transportation problem is classified as NP-hard [79], thus exact methods can only be applied to small instances and become prohibitive in time for real size problems. Due to the efficiency of the GA for solving complex and large size integer optimization problems, we apply it to our application to find near-optimal combinations in container delivery. Figure 5.7 demonstrates the general structure of GA. The explicit components of the developed genetic algorithm for solving the inland transportation problem are described in more details in the following subsections.

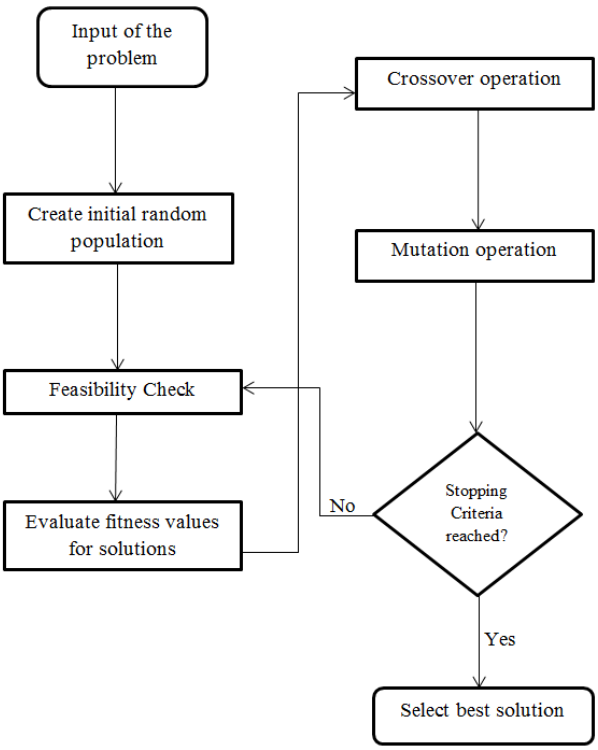

Figure 5.7: *Graph of the GA algorithm*

5.3.1 Chromosome representation of the problem

In this problem, the term *chromosome* is referred to as the sequence following which the order is to be considered for delivery. In this study, we use fixed length chromosomes which have the same length as the number of orders. Equivalently, the chromosome is defined as a permutation of all orders. Nevertheless, we have more than one truck and many order types; which truck is travelling which route, whether the chromosome is feasible are both important questions to answer. To deal with feasibility, we insert stoppers to the chromosome to break it into feasible and executable routes; each stopper means a new truck is allocated to perform the following task. To demonstrate the insertion process, we consider an example of 12 orders. The example also include two depots (D1, D2), single port (P) and 4 trucks (2×20ft, 2×40ft). As shown in Figure 5.8(A), a chromosome is a

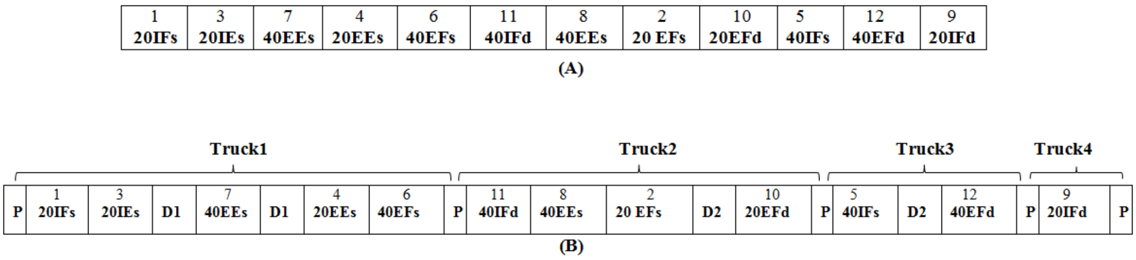

Figure 5.8: *Chromosome representation of Strip and Discharge with depot case*

random permutation of all orders. To make it an executable delivery plan, we insert the port (P) and depots (D1, D2) to the sequence so as to separate the chromosome into sub sequences, each representing a feasible route for a truck to travel (see Figure 5.8(B)). In details, the first truck (Truck1) starts from the port (P), taking a 20ft loaded *Strip* container to deliver it to location (1), then travelling to location (3) to collect a 20ft empty container. Since the next task is to deliver a 40ft empty container to location (7), the truck visits Depot (D1) to drop the 20ft empty on it and then collects a 40ft empty for the following order. This ”check and insert” process continues until accommodating order 6, when a 40ft loaded container that occupies the full capacity of the truck is collected and to be delivered to the port (where a route finishes). Since inserting any other orders after order 6 will create an infeasible route, a stopper (P) is then inserted to terminate the current route of the first truck. A new truck (route) is then considered to perform the rest of orders in the chromosome. The process is repeated until all feasible orders are considered for the chromosome (solution). In the end, when we have inserted all stoppers and depots into the chromosome, we can then examine each route to see which type of truck (20ft or 40ft) is needed to perform it so as to know the total number of trucks needed. It is obvious that using this insertion heuristic we can not pre-see the number of trucks we are going to use to execute all routes following the sequence as given in each chromosome.

Sometimes we may need less trucks than available but sometimes more. However, the only infeasibility that we can introduce for a chromosome is the larger number of trucks used, rather than creating completely un executable delivery plans, as if we include a certain number of routes/trucks and depot visiting options in the chromosome itself. Large number of infeasible solutions will slow down the convergence process of the GA, that is why we try to avoid infeasibility by running this insertion process to convert a sequence into executable solutions. The infeasibility in terms of extra trucks needed, e.g. we need three 40ft trucks in Figure 5.8 example but we just have 2, will then be penalised in the fitness function.

5.3.2 Constructing the initial population

To construct the initial population for the genetic algorithm, a number of chromosomes (solutions) are generated based on random permutation. As mentioned before, stoppers are used to ensure that all constructed routes are feasible. Note that we still denote the order permutation in figure 5.8(A) as the chromosome, rather than the extended executable delivery plan (Figure 5.8(B)). So although the number of trucks used and the number of nodes each truck is visiting are unpredictable, all chromosomes still have the same length.

5.3.3 Fitness value and evaluation

After constructing the initial population, the next step is to evaluate the quality of the chromosomes so as to rank them for future GA operations. The original objective of this problem is to minimize the total traveling cost to satisfy all orders. Nevertheless, the quality of a chromosome also depends on its feasibility. As said before, the only infeasibility one can expect after inserting all depots and stoppers is on the number of trucks. So in the fitness function we also add a term representing the penalty we have to pay for extra trucks needed. We define the fitness value (*Fit*) for solution (*i*) as:

$$Fit(i) = OBJ(i) + PENALTY(i) \tag{5.1}$$

$$= [TTD(i) + WTP(i)] + [PT20(i) + PT40(i)] \tag{5.2}$$

where $OBJ(i)$ is the original objective function which comprises: $TTD(i)$ as the total traveling distance of solution (*i*), $WTP(i)$ as the total penalty cost for violating the maximum allowed daily working time. $PENALTY(i)$ represents the penalty of violating the fleet size constraint, with $PT20(i)$ for 20ft trucks and $PT40(i)$ for 40ft trucks. Note that the $PT20$ and $PT40$ are set to very large values so as to avoid violating the fleet capacity constraints.

5.3.4 Selection process

The new population of chromosomes/solutions is created by performing certain recombination processes such as crossover and mutation. An appropriate selection process should be applied to choose parents. In this study, we apply the Roulette Wheel Selection (RWS) approach [77], which select parents according to their proportional performance; individuals with a smaller fitness values have a higher probability to be selected to generate offspring.

5.3.5 Genetic algorithm operators

After selecting solutions as parents from the initial population, a new population of offspring (children) could be constructed. Genetic operators represented by crossover and mutation are used to create the new populations as described later.

5.3.5.1 Crossover

The ″subschedule preservation crossover″, which was designed by Cheng and Gen [10] as a permutation based operator, is applied to construct new offspring. In this crossover method a sequence number of elitists are chosen randomly from the first parent to start the new children, then the remaining elitists to complete the first children are taken from the second parent without repetition. Similarly, to create the second children the process is started this time from the second parent. Note that we apply the subschedule preservation crossover on the preliminary constructed chromosomes which we demonstrated in Figure 5.8(A), since it is difficult to apply this method on the executable route in Figure 5.8(B). Figure 5.9 illustrates the crossover process for two chromosomes (parents). Firstly, orders 3, 5, 1 and 7 are selected from the first parent (Parent 1) to generate the first part of Child 1 as shown in Figure 5.9(A). To complete the chromosome (solution) of Child 1, the rest of orders are then taken from Parent 2 without any repetition for orders. In the same way, to construct the first part of Child 2, orders 5, 2, 6 and 3 are selected from Parent 2 and the rest of orders are collected from Parent 1 as in Figure 5.9(B).

Parent 1	3	5	1	7	9	6	4	2	8
Child 1	**3**	**5**	**1**	**7**	2	6	4	8	9
Parent 2	5	2	6	3	4	8	9	7	1

(A)

Parent 1	3	5	1	7	9	6	4	2	8
Child 2	**5**	**2**	**6**	**3**	1	7	9	4	8
Parent 2	5	2	6	3	4	8	9	7	1

(B)

Figure 5.9: *Crossover of orders*

5.3.5.2 Mutation

Mutation is a genetic algorithm operator which applies to allow more exploration for the search space and to avoid sticking at a local optimum. In the mutation process, two orders are selected randomly from the same chromosome (solution) which is also chosen randomly from the current population. The two orders are then swaped to obtain a new chromosome. As illustrated in Figure 5.10(A), Order 7 and Order 1 are selected randomly from a chromosome of 9 orders length. Then the two selected orders are exchanged to generate the new solution as given in Figure 5.10(B).

8	5	2	**7**	4	6	9	**1**	3

(A)

8	5	2	**1**	4	6	9	**7**	3

(B)

Figure 5.10: *Mutation of orders*

5.3.6 Stopping criterion

The above procedures of this genetic algorithm is repeated until the stopping criteria is reached. For this problem, the stopping criteria used is the maximum number of generations.

5.3.7 An example

To further demonstrate how the developed genetic algorithm approach can be applied and its performance, an example is given in this subsection. To benchmark the developed genetic algorithm, an upper bound can be obtained from the Mixed Integer Linear Programming model (MILP) for container inland transportation which we formulated in Chapter 4. The previous MILP model deals with order combinations for inland delivery under a similar context, say managing a single port and number of inland depots, planning the route for a number of orders including both loaded and empty ones using 20ft and 40ft fleets. The biggest difference is that this MILP model allows no more than four nodes to be visited on a single route, since it only considers *Strip* orders without an inland depot not *Street-turn* options. So no matter whether the container that is on the truck is loaded or empty, it has only one pair of (origin, destination) where exactly one end is fixed at the port.

This limits the transportation capacity of the truck and in the maximum case (40ft truck transporting 20ft orders), no more than four orders can be combined for the route. Therefore under the context of running without inland depot, we can use the MILP model to give us an upper bound on the exact cost of servicing the same group of *Strip* orders

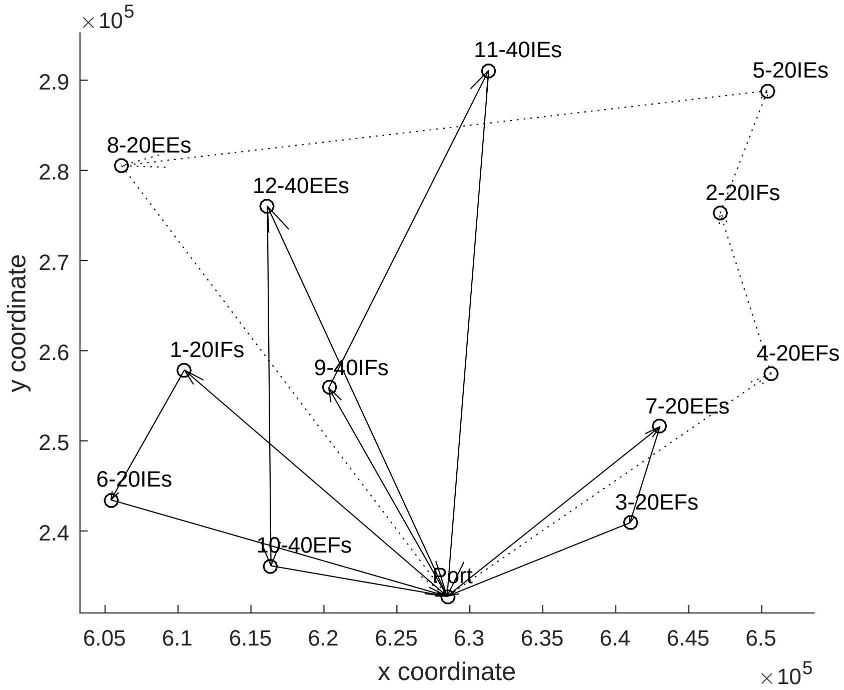

Figure 5.11: *Solution of the MILP model for Strip only without depot case*

using the same fleet, suppose the *Street-turn* is allowed so that more than four nodes can be visited.

Assume that a number of 12 orders of *Strip* type comprise: 2×20ft import loaded orders, 2 × 20ft import empty orders, 2 × 20ft export loaded orders, 2 × 20ft export empty orders, 1 × 40ft import loaded order, 1 × 40ft import empty order, 1 × 40ft export loaded order and 1 × 40ft export empty order. The available fleet sizes are 3 × 20ft trucks and 3 × 40ft trucks. Figure 5.11 illustrates the solution of the MILP model for this example. In this case, all orders are delivered jointly with others, creating one route combining 4 orders, four routes combining 2 orders. The total cost is 1175 and the number of the used trucks are 2 × 20ft trucks and 3 × 40ft trucks, respectively.

The same example is then solved by the genetic algorithm. The only modification that we have to make to the GA as given in Section 5.3.1 is for the insertion of an inland depot.

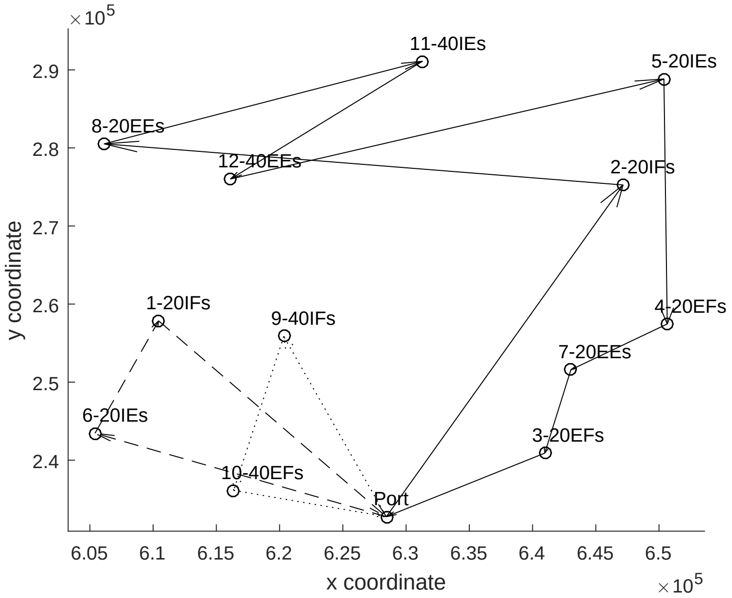

Figure 5.12: *Solution of the GA approach for Strip only without depot case*

Since an inland depot is no longer available in this example, upon any infeasible request based on empty containers, we initiate a new truck from the port. In this case all empty requests are matched perfectly using *Street-turn* and only 3×40ft trucks are used as shown in Figure 5.12. The total cost is 964. Note that this cost includes a penalty of 100/hour for the truck servicing eight orders, since it violated the maximum allowed working time by one hour. Nevertheless, paying the penalty is still worthwhile here as the total travelling distance is largely reduced.

To show how different settings can affect the solution and the benefit of allowing combined transportation of different types of containers, we consider the following three cases: *Strip* only with depot, *Discharge* only, *Strip* and *Discharge* with depot. Figure 5.13 shows the solution of the *Strip* only with depot case for the same set of the 12 orders that we have used for the *Strip* only without depot case, while this time we add inland depots so as to facilitate the *Depot-turn* option. Although, a penalty cost of 100/hour should be

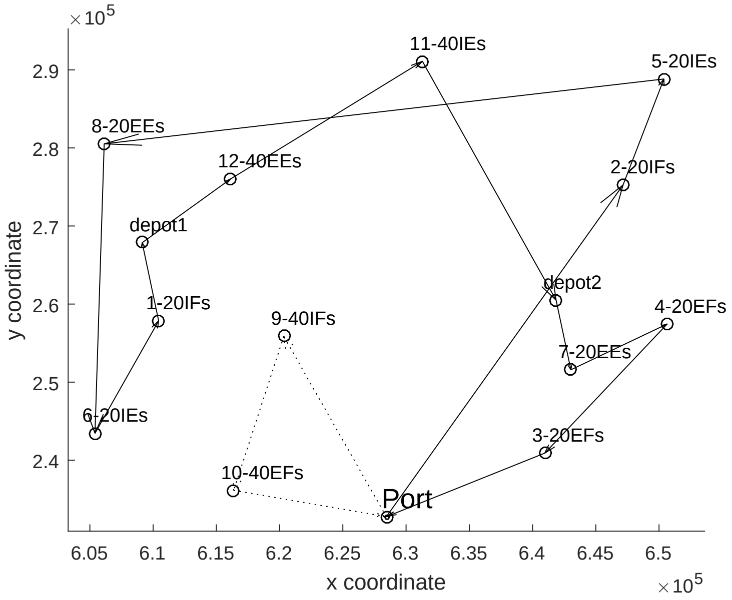

Figure 5.13: *Solution of the GA approach for Strip only with depot case*

paid for one of the trucks for violating the allowed working time, the total cost for this case is 903 which is less than for the *Strip* without depot case. The number of used trucks is only 2 × 40ft trucks, which is also less than for the *Strip* without depot case. Then, a set of 10 orders of discharge type for different locations is solved for the discharge only case as illustrated in Figure 5.14. In this case, the total cost is 1060 and the number of the used trucks for this case are 1 × 20ft trucks and 3 × 40ft trucks, respectively.

The set of orders for the two cases (12 *Strip* and 10 *Discharge*) are then aggregated to be solved by the general *Strip* and *Discharge* with depot case. As illustrated in Figure 5.15, there are 6 routes which require 6 × 40ft trucks. In this case the two depots (depot1, depot2) should be visited to drop/collect empty containers and the total cost for this case is 1708 which is less compared to the total cost of considering *Strip* and *Discharge* orders separately (1060 + 903 = 1963). This justifies our motivation of aggregating different types together.

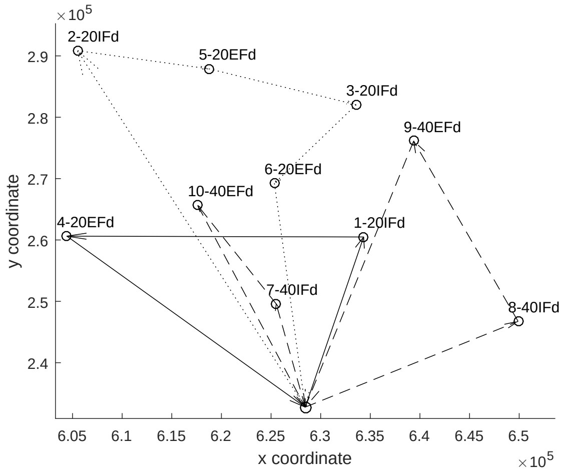

Figure 5.14: *Solution of the GA approach for Discharge only without depot case*

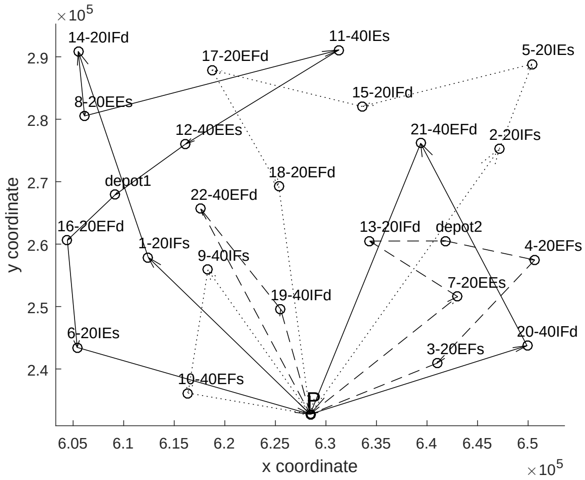

Figure 5.15: *Solution of the GA approach for the Strip and Discharge with depot case*

5.4 Computational experiments

A summary of some computational experiments is presented in this section for randomly generated instances. In these instances, depending on the size of the instances, geographical information is simulated according to the Port of Felixstowe, which is one of the largest container ports in the UK, and its major service areas distributed around the east of England. A shipping line, who own a number of 20ft and 40ft trucks is assumed to fulfil the transportation of orders between the port terminal, customers (importers and exporters) and a number of inland depots. Distances between the port and customers' locations and between each pair of customers are calculated based on the straight-line distance and rescaled to approximate the road distance. A penalty cost (200 pounds/h) applies for any extra working hours (more than 11 hours per day), according to the UK driving regulations. The service time at customer locations is approximated to 2 hours and the average speed for trucks is estimated at (40) mile/h. A varying number of orders of 20ft and 40ft loaded and empty containers are considered for delivery. All solution approaches (MILP, GA) are coded in MATLAB R2015b and executed on a CPU with an Intel(R)Core(TM)i7-4790 processor. The MILP is solved by CPLEX 12.6.1. For the GA, each example is solved 25 times to eliminate the impact of the random process. The reported result is an average of these 25 simulation runs. Other parameters of GA, such as the population sizes, number of generations, crossover and mutation rates, etc. are selected ad-hocly for every example.

5.4.1 Results for *Strip* only without depot case using the MILP and GA

In this section we test to see the gap of the developed GA approach from the upper bound solution obtained from solving the MILP model (formulated in Chapter 4) for small size instances. As shown in Table 5.2, the MILP and the GA are tested for different sizes of instances with 12, 20, 50, 70 and 100 orders. For each instance, the number of orders for each type of container is given in the second and third column of Table 5.2, in the form of (IF, EF, IE, EE). The result shows that for all instances the GA improves the MILP by 8-13% in the operations cost of the final delivery plan. The number of the used trucks for the GA is also less than the number required by the MILP solution. Note that the

#	Number and Type of orders		# fleet				(Average of) penalty	(Average of)	Standard	CPU	(%)
Orders	# 20ft	# 40ft	available		average used		for extra	total cost	deviation	time(sec.)	Improvement
	Strip	Strip	20ft	40ft	20ft	40ft	working time				on MILP
MILP solution for Strip only without depot											
12	(2,2,2,2)	(1,1,1,1)	3	3	2	3	-	1175	-	00:16	-
20	(3,3,3,3)	(2,2,2,2)	5	5	4	5	-	2307	-	00:06	-
50	(8,8,8,8)	(5,5,4,4)	15	20	-	18	200	5060	-	43:63	-
70	(10,10,10,10)	(8,7,8,7)	25	35	-	25	1200	8949	-	358:23	-
100	(15,15,10,10)	(13,12,13,12)	35	45	-	40	600	10778	-	2885:33	-
GA solution for Strip only without depot											
12	(2,2,2,2)	(1,1,1,1)	3	3	0.08	2.28	200	1064.8	35.494	02:46	9%
20	(3,3,3,3)	(2,2,2,2)	5	5	0.76	4.04	392	2102.2	85.171	03:08	9%
50	(8,8,8,8)	(5,5,4,4)	15	20	0.48	12.76	432	4644.4	191.24	05:05	8%
70	(10,10,10,10)	(8,7,8,7)	25	35	0.52	19.08	1144	7753.8	254.57	07:15	13%
100	(15,15,10,10)	(13,12,13,12)	35	45	0.68	30.12	848	9377.6	404.27	10:18	13%

Table 5.2: *Results for solving small examples by MILP and GA.*

number of the used trucks is fractional in the GA solution as reported in the table, since we run the GA for 25 times and took the average. Moreover, the computational times for the GA are just a few seconds which are much smaller than the solution times of the MILP for instances larger than 50. From the percentage improvement of GA model on the MILP costs, we can see that the GA performs well as a solution approach, which provides solutions well below the theoretical upper bounds (MILP) for small instances. Note that the MILP solution (exact optimal) is a theoretical upper bound of the true optimum of the problem solved by the GA, as the MILP allows no more than 4 orders on each route which restricts allowing the empty movements with *Street-turn*. Inland depots have been ruled out by the problem settings. We compare the GA solution with this upper bound simply because an exact model under the same problem settings is not available through current literature. Amongst those models that are available, the MILP model (formulated in Chapter 4) is the one that solves largest size instances in affordable computation time. So despite there are gaps in the solution caused both by the problem settings and by the

GA as a meta-heuristic approach, the comparison still informs us about how good the designed GA is in solving this type of problem to some extent. This actually shows a common situation in practice. When a problem is too difficult to solve to optimality, one can choose to simplify the modelling or to simplify the solution approach (like use GA to solve the non-restricted situation). Suppose by doing the latter is justified better than doing the former (like restrict to 4 orders to define the MILP model to allow exact solution), we still contribute to the problem by directing a correct way to tackle it.

5.4.2 The value of combining *Strip* and *Discharge* orders

In this section, we investigate how much one can save by combining *Strip* orders with *Discharge* ones. To achieve this aim, we use GA to solve three problem settings: *Strip* only with depot, *Discharge* only and the *Strip* and *Discharge* with depot cases. In Table 5.3 we summarize the results for instances with 25, 50, 250, 500 and 1000 orders, solved separately and jointly. We can see from Figures 5.16, 5.17 that the GA for the *Strip* and *Discharge* only cases is not improving after running for about 300 generations. While for the joint case it needs about 800 generations as shown in Figure 5.18. As illustrated in Table 5.3,

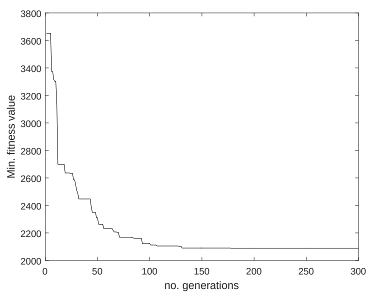

Figure 5.16: *GA performance for Strip only*

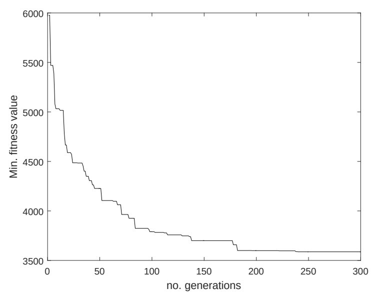

Figure 5.17: *GA performance for Discharge only*

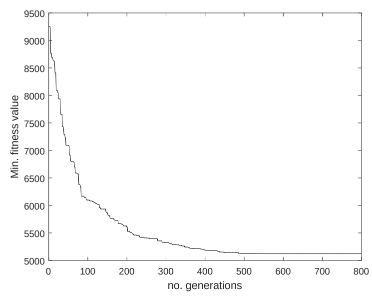

Figure 5.18: *GA performance for Joint case*

the result shows that the average cost is reduced by 6-16% than the summation of the two

# Orders	Number and Type of orders				# fleet				Penalty of extra working time	Mean of total cost	Stand. dev.	CPU time(sec.)	Saving cost(%) from separate cases
	# 20ft	# 40ft	# 20ft	# 40ft	available		average used						
	Strip	Strip	Disch.	Disch.	20ft	40ft	20ft	40ft					
25-Strip	(4,4,4,3)	(3,3,2,2)	-	-	5	10	0.3	6.2	228	1923.6	97.4	02:32	-
25-Disch.	-	-	(8,7)	(5,5)	5	10	1.1	9.04	316	3214.3	55.47	02:25	-
50-Separate	**(4,4,4,3)**	**(3,3,2,2)**	**(8,7)**	**(5,5)**	**10**	**20**	**1.4**	**15.24**	**544**	**5137.9**	**152.87**	**04:57**	
50-Joint	**(4,4,4,3)**	**(3,3,2,2)**	**(8,7)**	**(5,5)**	**10**	**20**	**0.4**	**15.04**	**508**	**4811.9**	**124.9**	**41:94**	**6%**
50-Strip	(8,8,7,7)	(5,5,5,5)	-	-	10	20	0.52	11.84	484	4049.4	154.7	03:64	-
50-Disch.	-	-	(15,15)	(10,10)	10	20	2	19.24	716	6957.8	133.61	03:48	-
100-Separate	**(8,8,7,7)**	**(5,5,5,5)**	**(15,15)**	**(10,10)**	**20**	**40**	**2.52**	**31.08**	**1200**	**11007.2**	**288.31**	**07:12**	
100-Joint	**(8,8,7,7)**	**(5,5,5,5)**	**(15,15)**	**(10,10)**	**20**	**40**	**0.9**	**30.3**	**1116**	**9802.2**	**218.12**	**68:44**	**11%**
250-Strip	(40,40,35,35)	(25,25,25,25)	-	-	50	110	5.92	68.48	3304	27233	558.73	09:35	-
250-Disch.	-	-	(75,75)	(50,50)	50	110	23.6	108.28	4532	43280	621.64	09:12	-
500-Separate	**(40,40,35,35)**	**(25,25,25,25)**	**(75,75)**	**(50,50)**	**100**	**220**	**29.52**	**176.76**	**7836**	**70513**	**1180.37**	**18:47**	
500-Joint	**(40,40,35,35)**	**(25,25,25,25)**	**(75,75)**	**(50,50)**	**100**	**220**	**9.68**	**169.16**	**6732**	**59554**	**773.36**	**273:64**	**16%**
500-Strip	(80,80,70,70)	(50,50,50,50)	-	-	100	250	12.56	147	7792	61421	834.22	18:50	-
500-Disch.	-	-	(150,150)	(100, 100)	100	250	45.92	244.68	9184	91515	1315.9	18:08	-
1000-Separate	**(80,80,70,70)**	**(50,50,50,50)**	**(150,150)**	**(100,100)**	**200**	**500**	**58.48**	**391.68**	**16976**	**152936**	**2150.12**	**36:58**	
1000-Joint	**(80,80,70,70)**	**(50,50,50,50)**	**(150,150)**	**(100,100)**	**200**	**500**	**26.28**	**360.64**	**15084**	**131616**	**1114.5**	**532:88**	**14%**

Table 5.3: *Results for combining the* Strip *and* Discharge *orders with depot case*

subproblems obtained by disaggregating over order type, which is the common type of research that has been established in container delivery. The smaller overall cost is not just obtained by saving travelling distance, but also through shorter over-time working of truck drivers which reduces the paid penalty cost. The number of used trucks is also reduced for the joint case compared to the separate case. For instance, in the example of 1000 orders, servicing the *Strip* and *Discharge* orders separately needs 58 20ft trucks and 392 40ft trucks. While for the joint case we just need 26 20ft and 361 40ft trucks, which means that in total 60 (trucks and drivers) will be saved from working. However, the computational time to perform the joint case is larger (still acceptable) compared to the two individual cases, since that larger number of orders is solved jointly.

# Orders	Number and Type of orders				# fleet				Penalty of extra working time	Mean of total cost	Stand. dev.	CPU time(sec.)	Cost (%) worse than with depot case
	# 20ft Strip	# 40ft Strip	# 20ft Disch.	# 40ft Disch.	available		average used						
					20ft	40ft	20ft	40ft					
50-with depot	(4,4,4,3)	(3,3,2,2)	(8,7)	(5,5)	10	20	0.4	15.04	508	4811.9	124.9	41:94	
50-without	(4,4,4,3)	(3,3,2,2)	(8,7)	(5,5)	10	20	0.32	15.52	572	5117.2	118.69	35:04	6%
100-with depot	(8,8,7,7)	(5,5,5,5)	(15,15)	(10,10)	20	40	0.92	30.28	1116	9802.2	218.12	68:44	
100-without	(8,8,7,7)	(5,5,5,5)	(15,15)	(10,10)	20	40	0.8	31.56	1216	10629	189.82	54:64	8%
500-with depot	(40,40,35,35)	(25,25,25,25)	(75,75)	(50,50)	100	220	9.68	169.16	6732	59554	773.36	273:64	
500-without	(40,40,35,35)	(25,25,25,25)	(75,75)	(50,50)	100	220	17.04	196.2	7244	67571	1034.4	231:33	12%
1000-with depot	(80,80,70,70)	(50,50,50,50)	(150,150)	(100, 100)	200	500	26.28	360.64	15084	131616	1114.5	532:88	
1000-without	(80,80,70,70)	(50,50,50,50)	(150,150)	(100, 100)	200	500	50.68	437.12	15836	149941	1232.7	461:36	12%

Table 5.4: *Results for comparing the* Strip *and* Discharge *case, with and without depots.*

5.4.3 The value of the inland depots

In this section, we will evaluate the value of using the inland depots in the inland transportation. For this purpose, we test the genetic algorithm for the *Strip* and *Discharge* without depot case. The same data for the *Strip* and *Discharge* with depot case is used, but this time without the existence once of inland depots. As shown in Table 5.4, the average total cost is increased by 6-12% for the inclusion of depots. Indeed, without depots we need more trucks to satisfy all orders than if there is a depot. It is clear that the use of inland depots has major benefits. The reduced overall cost is obtained from better routes that can be constructed as more flexible combinations are allowed, and a higher truck utilization.

5.5 Summary

In this chapter the transportation of two types of orders: 20ft and 40ft loaded and empty containers using a fleet of 20ft and 40ft long trucks is investigated. A Genetic Algorithm

(GA) approach is developed for this purpose. Both cases of the transportation which are the *Strip* and *Discharge* of containers have been considered in this research. The two major strategies for storing and relocating empty containers which are the *Street-turn* and *Depot-turn* cases are also considered. These cases are investigated separately and jointly. The result shows that:

1. GA is a better approach than solving the simplified problem using an exact method.
2. It is worthwhile to combine *Strip* and *Discharge* orders in delivery.
3. Inland empty depots help saving delivery cost by allowing more flexible delivery routes and improving truck utilization.

Chapter 6

Conclusions and future work

This chapter is in two parts. The first part presents the main conclusions whereas the second part highlights future work and directions for worthwhile research.

6.1 Main conclusions

The inland transportation takes a significant portion of the total cost that arises from intermodal transportation. In addition, there are many parties (shipping lines, haulage companies, customers) who share this operation as well as many restrictions that increase the complexity of this problem and make it NP-hard. Therefore, it is important to create an efficient strategy to manage this process in a way to ensure all parties are satisfied.

In Chapter 2, a comprehensive literature review of pre existing literature of container inland (drayage) transportation is presented. Some studies consider the movement of one individual container per truck, of the size 20/40ft, while some others focus upon the delivery of one 40ft and the pairing of two 20ft containers per truck. Articles are also reviewed based on the stripe and discharge category of containers. Methodology of the research is illustrated in this chapter.

In Chapter 3, the Pairing of Containers/Orders in Drayage Transportation (PCDT) from the perspective of delivering paired containers on 40ft truck and/or individual containers on 20ft truck, between a single port and a list of customer locations is investigated. An assignment Mixed Integer Linear Programming (MILP) model is formulated, which solves the problem of how to combine orders in delivery to save the total transportation cost when orders with both single and multiple destinations exist. In opposition to the traditional models relying on the Vehicle Routing Problem with Simultaneous Pickups and Deliveries and Time Windows (VRP-SPDTW) formulation, this model falls into the assignment problem category which is more efficient to solve on large size instances. Another merit for the proposed model is that it can be implemented on different variants of the container drayage problem: import only, import-inland and import-inland-export. Results show that in all cases the pairing of containers yields less cost compared to the individual delivery and decreases empty tours. The proposed model can be solved to optimality efficiently (within half an hour) for over 300 orders.

In Chapter 4, a Mixed Integer Linear Programming (MILP) model is designed for combining orders in the inland, haulage transportation of containers. In this MILP model, the pick up and delivery process of both 20 and 40 foot containers from the terminals to the customer locations and vice versa are optimized using heterogenous fleet consisting of both 20ft and 40ft trucks/chasses. Important operational constraints such as the time window at order receivers, the payload weight of containers and the regulation of the working hours are considered. Based on an assignment model, this MILP solves problems with 100 orders efficiently to optimality.

To deal with larger instances, a decomposition and aggregation heuristic is designed. The basic idea of this approach is to decompose order locations geographically into fan-shaped sub-areas based on the angle of the order location to the port, and solve the sub problems using the proposed MILP model. To balance the fleet size amongst all subgroups, column generation is used to iteratively adjust the number of allocated trucks according to the shadow-price of each truck type. Based on decomposed solutions, orders that are "fully" combined with others are removed and an aggregation phase follows to enable wider combination choices across subgroups. The decomposition and aggregation solution process is tested to be both efficient and cost-saving.

In Chapter 5, we reflect the real practice of haulage container transportation where both pick-up and delivery, empty and loaded, *Discharge* and *Strip* of heterogenous container types are combined. Heterogenous fleets are considered to perform the inland transportation. For managing the delivery of empty containers, the two common strategies, *Depot-turn* and *Street-turn* are both tested with examples capturing real geographical information. A Genetic Algorithm (GA) approach is designed for solving large scale problems. The result shows that solving the complicated problem using the developed GA is better than solving the simplified problem using an exact method, even on small scale instances where an optimal solution is achievable with the exact model. In addition, combining the *Strip* and *Discharge* types with the usage of inland empty depots both saves transportation costs and increases fleet utilization. The value of using inland depots is also evaluated.

6.2 Suggestions for further work

The large number of decision variables and constraints included in the inland transportation of containers models prohibits the exact mathematical models in this thesis and other literature from solving larger size problems. Thus, creating and developing new solution techniques such as exploiting structural property and hybrid methods to obtain the exact solution will be difficult and challenging for future research.

In this thesis, we assumed that information and data of the studied problems are deterministic and known in advance. However, these assumptions are sometimes unsuitable in reality because of uncertainty and ambiguity. Uncertainty for container transportation usually exists in cases such as changing the time windows of orders, the late arrival time of a vessel to the port, the breakdown of trucks and traffic congestion. Therefore, stochastic and dynamic optimization techniques are needed to tackle uncertainty and help to make the best decisions in the container delivery industry.

Here, we investigate the delivery of container inland transportation from the perspective of single objective function mathematical models, however, usually companies target to achieve more than one objective at the same time such as minimizing the delivery cost, maximizing truck utilization, minimizing waiting time at customer locations, minimizing idle time and minimizing the number of drivers required for the service,...etc. Therefore, this calls upon the use of multi-objective optimization to achieve several aims simultaneously.

Bibliography

[1] R. Battiti and G. Tecchiolli. The reactive tabu search. *ORSA journal on computing*, 6(2):126–140, 1994.

[2] L. Bodin, A. Mingozzi, R. Baldacci, and M. Ball. The rollon–rolloff vehicle routing problem. *Transportation Science*, 34(3):271–288, 2000.

[3] K. Braekers, A. Caris, and G. K. Janssens. Integrated planning of loaded and empty container movements. *OR spectrum*, pages 1–22, 2013.

[4] K. Braekers, A. Caris, and G. K. Janssens. Bi-objective optimization of drayage operations in the service area of intermodal terminals. *Transportation Research Part E: Logistics and Transportation Review*, 65:50–69, 2014.

[5] K. Braekers, G. K. Janssens, and A. Caris. Challenges in managing empty container movements at multiple planning levels. *Transport Reviews*, 31(6):681–708, 2011.

[6] J. Bramel and D. Simchi-Levi. A location based heuristic for general routing problems. *Operations research*, 43(4):649–660, 1995.

[7] C. Caballini, I. Rebecchi, and S. Sacone. Combining multiple trips in a port environment for empty movements minimization. *Transportation Research Procedia*, 10:694–703, 2015.

[8] A. Caris and G. K. Janssens. A local search heuristic for the pre-and end-haulage of intermodal container terminals. *Computers & Operations Research*, 36(10):2763–2772, 2009.

[9] A. Caris and G. K. Janssens. A deterministic annealing algorithm for the pre-and end-haulage of intermodal container terminals. *International Journal of Computer Aided Engineering and Technology*, 2(4):340–355, 2010.

[10] R. Cheng and M. Gen. Parallel machine scheduling problems using memetic algorithms. *Computers & Industrial Engineering*, 33(3-4):761–764, 1997.

[11] R. K. Cheung, N. Shi, W. B. Powell, and H. P. Simao. An attribute–decision model for cross-border drayage problem. *Transportation Research Part E: Logistics and Transportation Review*, 44(2):217–234, 2008.

[12] H. R. Choi, B. K. Park, J. Lee, and C. Park. Dispatching of container trucks using genetic algorithm. In *Interaction Sciences (ICIS), 2011 4th International Conference on*, pages 146–151. IEEE, 2011.

[13] M. Christiansen, K. Fagerholt, B. Nygreen, and D. Ronen. Maritime transportation. *Handbooks in operations research and management science*, 14:189–284, 2007.

[14] M. Christiansen, K. Fagerholt, and D. Ronen. Ship routing and scheduling: Status and perspectives. *Transportation science*, 38(1):1–18, 2004.

[15] N. Christofides. The vehicle routing problem. *Revue française d'automatique, informatique, recherche opérationnelle. Recherche opérationnelle*, 10(V1):55–70, 1976.

[16] K. H. Chung, C. S. Ko, J. Y. Shin, H. Hwang, and K. H. Kim. Development of mathematical models for the container road transportation in korean trucking industries. *Computers & Industrial Engineering*, 53(2):252–262, 2007.

[17] G. Clarke and J. W. Wright. Scheduling of vehicles from a central depot to a number of delivery points. *Operations research*, 12(4):568–581, 1964.

[18] L. Coslovich, R. Pesenti, and W. Ukovich. Minimizing fleet operating costs for a container transportation company. *European Journal of Operational Research*, 171(3):776–786, 2006.

[19] A. J. Craig, E. E. Blanco, and Y. Sheffi. Estimating the co 2 intensity of intermodal freight transportation. *Transportation Research Part D: Transport and Environment*, 22:49–53, 2013.

[20] T. G. Crainic and K. H. Kim. Intermodal transportation. *Handbooks in operations research and management science*, 14:467–537, 2007.

[21] C. F. Daganzo. The distance traveled to visit n points with a maximum of c stops per vehicle: An analytic model and an application. *Transportation Science*, 18(4):331–350, 1984.

[22] C. F. Daganzo. The length of tours in zones of different shapes. *Transportation Research Part B: Methodological*, 18(2):135–145, 1984.

[23] H. A. Daham, X. Yang, and M. K. Warnes. An efficient mixed integer programming model for pairing containers in inland transportation based on the assignment of orders. *Journal of the Operational Research Society*, 68(6):678–694, 2017.

[24] G. B. Dantzig and P. Wolfe. Decomposition principle for linear programs. *Operations research*, 8(1):101–111, 1960.

[25] L. Davis. Handbook of genetic algorithms. 1991.

[26] D. de Werra and A. Hertz. Tabu search techniques. *Operations-Research-Spektrum*, 11(3):131–141, 1989.

[27] J. Desrosiers and M. E. Lübbecke. A primer in column generation. In *Column generation*, pages 1–32. Springer, 2005.

[28] H. A. Eiselt and C.-L. Sandblom. *Linear programming and its applications*. Springer Science & Business Media, 2007.

[29] Z. Fang, W. Tu, Q. Li, S.-L. Shaw, S. Chen, and B. Y. Chen. A voronoi neighborhood-based search heuristic for distance/capacity constrained very large vehicle routing problems. *International Journal of Geographical Information Science*, 27(4):741–764, 2013.

[30] S. S. Fazi. *Mode selection, routing and scheduling for inland container transport*. PhD thesis, Technische Universiteit Eindhoven, 2014.

[31] M. L. Fisher and R. Jaikumar. A generalized assignment heuristic for vehicle routing. *Networks*, 11(2):109–124, 1981.

[32] L. R. Ford Jr and D. R. Fulkerson. A suggested computation for maximal multi-commodity network flows. *Management Science*, 5(1):97–101, 1958.

[33] P. Francis, G. Zhang, and K. Smilowitz. Improved modeling and solution methods for the multi-resource routing problem. *European Journal of Operational Research*, 180(3):1045–1059, 2007.

[34] J. Funke and H. Kopfer. A neighborhood search for a multi-size container transportation problem. *IFAC-PapersOnLine*, 48(3):2041–2046, 2015.

[35] J. Funke and H. Kopfer. Container flows and truck routes in inland container transportation. In *Dynamics in Logistics*, pages 215–225. Springer, 2016.

[36] J. Funke and H. Kopfer. A model for a multi-size inland container transportation problem. *Transportation Research Part E: Logistics and Transportation Review*, 89:70–85, 2016.

[37] M. Gendreau and J.-Y. Potvin. Metaheuristics in combinatorial optimization. *Annals of Operations Research*, 140(1):189–213, 2005.

[38] A. Ghezelsoflu, M. Di Francesco, A. Frangioni, and P. Zuddas. A set-covering formulation for a drayage problem with single and double container loads. *Journal of Industrial Engineering International*, pages 1–12, 2018.

[39] B. E. Gillett and L. R. Miller. A heuristic algorithm for the vehicle-dispatch problem. *Operations research*, 22(2):340–349, 1974.

[40] P. C. Gilmore and R. E. Gomory. A linear programming approach to the cutting-stock problem. *Operations research*, 9(6):849–859, 1961.

[41] F. Glover. Heuristics for integer programming using surrogate constraints. *Decision Sciences*, 8(1):156–166, 1977.

[42] F. Glover. Future paths for integer programming and links to artificial intelligence. *Computers & operations research*, 13(5):533–549, 1986.

[43] F. Glover and H. J. Greenberg. New approaches for heuristic search: A bilateral linkage with artificial intelligence. *European Journal of Operational Research*, 39(2):119–130, 1989.

[44] M. Gronalt, R. F. Hartl, and M. Reimann. New savings based algorithms for time constrained pickup and delivery of full truckloads. *European Journal of Operational Research*, 151(3):520–535, 2003.

[45] D. Hajem, A. and X. Yang. A column generation based decomposition approach for combining orders in inland transportation of containers. *OR Spectrum Journal*, Submitted on 12/07/2018.

[46] D. Hajem, A. and X. Yang. A genetic algorithm approach for the combined stripe and discharge delivery of containers. *Computers & Industrial Engineering Journal*, Submitted on 07/08/2018.

[47] P. Hansen and N. Mladenović. An introduction to variable neighborhood search. In *Meta-heuristics*, pages 433–458. Springer, 1999.

[48] P. Hansen and N. Mladenović. Variable neighborhood search: Principles and applications. *European journal of operational research*, 130(3):449–467, 2001.

[49] R. Harrison, N. Hutson, J. West, and J. Wilke. Characteristics of drayage operations at the port of houston. Technical report, Southwest Region University Transportation Center, Center for Transportation Research, University of Texas at Austin, 2008.

[50] J. Holland. Adaptation in natural and artificial systems. mi. *Ann Arbor: University of Michigan Press*, 1975.

[51] M. I. Hosny. *Investigating heuristic and meta-heuristic algorithms for solving pickup and delivery problems*. Cardiff University, 2010.

[52] Y. Ileri, M. Bazaraa, T. Gifford, G. Nemhauser, J. Sokol, and E. Wikum. An optimization approach for planning daily drayage operations. *Central European Journal of Operations Research*, 14(2):141–156, 2006.

[53] A. Imai, E. Nishimura, and J. Current. A lagrangian relaxation-based heuristic for the vehicle routing with full container load. *European journal of operational research*, 176(1):87–105, 2007.

[54] J.-J. Jaw, A. R. Odoni, H. N. Psaraftis, and N. H. Wilson. A heuristic algorithm for the multi-vehicle advance request dial-a-ride problem with time windows. *Transportation Research Part B: Methodological*, 20(3):243–257, 1986.

[55] H. Jula, A. Chassiakos, and P. Ioannou. Port dynamic empty container reuse. *Transportation Research Part E: Logistics and Transportation Review*, 42(1):43–60, 2006.

[56] H. Jula, M. Dessouky, P. Ioannou, and A. Chassiakos. Container movement by trucks in metropolitan networks: modeling and optimization. *Transportation Research Part E: Logistics and Transportation Review*, 41(3):235–259, 2005.

[57] A. Khachaturyan, S. Semenovskaya, and B. Vainstein. A statistical-thermodynamic approach to determination of structure amplitude phases. *Sov. Phys. Crystallogr*, 24:519–524, 1979.

[58] A. Khachaturyan, S. Semenovsovskaya, and B. Vainshtein. The thermodynamic approach to the structure analysis of crystals. *Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography*, 37(5):742–754, 1981.

[59] S. Kirkpatrick, C. Gelatt Jr, and M. Vecchi. Optimization by simulated annealing. *SCIENCE*, 220(4598), 1983.

[60] R. Konings. Foldable containers to reduce the costs of empty transport? a cost–benefit analysis from a chain and multi-actor perspective. *Maritime Economics & Logistics*, 7(3):223–249, 2005.

[61] H. Kopfer, J. Schönberger, S. Sterzik, and R. Zhang. Optimization of inland container transportation with and without container sharing. In *Proceedings of the 1st International Conference on Logistics and Maritime Systems*, 2010.

[62] M. Lai. *Models and algorithms for the empty container repositioning and its integration with routing problems*. PhD thesis, Universita'degli Studi di Cagliari, 2013.

[63] M. Lai, M. Battarra, M. Di Francesco, and P. Zuddas. An adaptive guidance metaheuristic for the vehicle routing problem with splits and clustered backhauls. *Journal of the Operational Research society*, 66(7):1222–1235, 2015.

[64] M. Lai, T. G. Crainic, M. Di Francesco, and P. Zuddas. An heuristic search for the routing of heterogeneous trucks with single and double container loads. *Transportation Research Part E: Logistics and Transportation Review*, 56:108–118, 2013.

[65] A. H. Land and A. G. Doig. An automatic method of solving discrete programming problems. *Econometrica: Journal of the Econometric Society*, pages 497–520, 1960.

[66] M. Levinson. *The Box: How the Shipping Container Made the World Smaller and the World Economy Bigger-with a new chapter by the author*. Princeton University Press, 2016.

[67] Y. V. Lun, K.-h. Lai, and T. E. Cheng. *Shipping and logistics management*. Springer, 2010.

[68] C. Macharis and Y. M. Bontekoning. Opportunities for or in intermodal freight transport research: A review. *European Journal of operational research*, 153(2):400–416, 2004.

[69] R. Mole and S. Jameson. A sequential route-building algorithm employing a generalised savings criterion. *Journal of the Operational Research Society*, 27(2):503–511, 1976.

[70] E. K. Morlok and L. N. Spasovic. Redesigning rail-truck intermodal drayage operations for enhanced service and cost performance. In *Journal of the Transportation Research Forum*, volume 34, 1994.

[71] P. Nagl. *Longer combination vehicles (LCV) for Asia and the Pacific region: some economic implications*. United Nations Publications, 2007.

[72] R. Namboothiri and A. L. Erera. Planning local container drayage operations given a port access appointment system. *Transportation Research Part E: Logistics and Transportation Review*, 44(2):185–202, 2008.

[73] N. Nordsieck, T. Buer, and J. Schönberger. Potential of improving truck-based drayage operations of marine terminals through street turns. In *Dynamics in Logistics*, pages 433–443. Springer, 2017.

[74] J. Nossack and E. Pesch. A truck scheduling problem arising in intermodal container transportation. *European Journal of Operational Research*, 230(3):666–680, 2013.

[75] T. E. Notteboom* and J.-P. Rodrigue. Port regionalization: towards a new phase in port development. *Maritime Policy & Management*, 32(3):297–313, 2005.

[76] M. Padberg and G. Rinaldi. A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems. *SIAM review*, 33(1):60–100, 1991.

[77] H. M. Pandey. Performance evaluation of selection methods of genetic algorithm and network security concerns. *Procedia Computer Science*, 78:13–18, 2016.

[78] S. Parragh, K. Doerner, and R. Hartl. A survey on pickup and delivery problems: Part 1: Transportation between customers and depot and part 2: Transportation between pickup and delivery locations. *Journal für Betriebswirtschaft*, 58(1):21–51, 2008.

[79] D. Popović, M. Vidović, and M. Nikolić. The variable neighborhood search heuristic for the containers drayage problem with time windows. In *Soft Computing in Industrial Applications*, pages 351–364. Springer, 2014.

[80] U. N. publication. Review of maritime transport. *New York and Geneva*, Sales No. E.15.II.D.6, 2015.

[81] C. R. Reeves. Genetic algorithms for the operations researcher. *INFORMS journal on computing*, 9(3):231–250, 1997.

[82] L. Reinhardt, S. Spoorendonk, and D. Pisinger. Solving vehicle routing with full container load and time windows. *Computational Logistics*, pages 120–128, 2012.

[83] L. B. Reinhardt, D. Pisinger, S. Spoorendonk, and M. M. Sigurd. Optimization of the drayage problem using exact methods. *INFOR: Information Systems and Operational Research*, 54(1):33–51, 2016.

[84] J.-P. Rodrigue, C. Comtois, and B. Slack. *The geography of transport systems*. Routledge, 2016.

[85] J. Schönberger, T. Buer, and H. Kopfer. A model for the coordination of 20-foot and 40-foot container movements in the hinterland of a container terminal. In *International Conference on Computational Logistics*, pages 113–127. Springer, 2013.

[86] F. Schulte, E. Lalla-Ruiz, R. G. González-Ramírez, and S. Voß. Reducing port-related empty truck emissions: a mathematical approach for truck appointments with collaboration. *Transportation Research Part E: Logistics and Transportation Review*, 105:195–212, 2017.

[87] K. Shin and S. Han. A centroid-based heuristic algorithm for the capacitated vehicle routing problem. *Computing and Informatics*, 30(4):721–732, 2012.

[88] S. Shiri and N. Huynh. Optimization of drayage operations with time-window constraints. *International Journal of Production Economics*, 176:7–20, 2016.

[89] S. Shiri and N. Huynh. Assessment of us chassis supply models on drayage productivity and air emissions. *Transportation Research Part D: Transport and Environment*, 2017.

[90] K. Smilowitz. Multi-resource routing with flexible tasks: an application in drayage operations. *Iie Transactions*, 38(7):577–590, 2006.

[91] Y. Song, J. Zhang, Z. Liang, and C. Ye. An exact algorithm for the container drayage problem under a separation mode. *Transportation Research Part E: Logistics and Transportation Review*, 106:231–254, 2017.

[92] R. Stahlbock and S. Voß. Operations research at container terminals: a literature update. *OR spectrum*, 30(1):1–52, 2008.

[93] D. Steenken, S. Voß, and R. Stahlbock. Container terminal operation and operations research-a classification and literature review. *OR spectrum*, 26(1):3–49, 2004.

[94] S. Sterzik and H. Kopfer. A tabu search heuristic for the inland container transportation problem. *Computers & Operations Research*, 40(4):953–962, 2013.

[95] S. Sterzik, H. Kopfer, and J. Funke. Advantages of decoupling containers and vehicles at customer locations. In *Logistics Management*, pages 301–312. Springer, 2015.

[96] K. C. Tan, Y. Chew, and L. H. Lee. A hybrid multi-objective evolutionary algorithm for solving truck and trailer vehicle routing problems. *European Journal of Operational Research*, 172(3):855–885, 2006.

[97] M. Vidović, M. Nikolić, and D. Popović. Two mathematical formulations for the containers drayage problem with time windows. In *2nd International Conference on Supply Chains ICSC*, 2012.

[98] M. Vidović, D. Popović, B. Ratković, and G. Radivojević. Generalized mixed integer and vns heuristic approach to solving the multisize containers drayage problem. *International Transactions in Operational Research*, 24(3):583–614, 2017.

[99] M. Vidović, G. Radivojević, and B. Raković. Vehicle routing in containers pickup up and delivery processes. *Procedia-Social and Behavioral Sciences*, 20:335–343, 2011.

[100] W. F. Wang and W. Y. Yun. Scheduling for inland container truck and train transportation. *International journal of production economics*, 143(2):349–356, 2013.

[101] X. Wang and A. C. Regan. Local truckload pickup and delivery with hard time window constraints. *Transportation Research Part B: Methodological*, 36(2):97–112, 2002.

[102] S. Wen and P. Zhou. A container vehicle routing model with variable traveling time. In *Automation and Logistics, 2007 IEEE International Conference on*, pages 2243–2247. IEEE, 2007.

[103] A. Wren and A. Holliday. Computer scheduling of vehicles from one or more depots to a number of delivery points. *Operational Research Quarterly*, pages 333–344, 1972.

[104] Z. Xue, W.-H. Lin, L. Miao, and C. Zhang. Local container drayage problem with tractor and trailer operating in separable mode. *Flexible Services and Manufacturing Journal*, 27(2-3):431–450, 2015.

[105] Z. Xue, C. Zhang, W.-H. Lin, L. Miao, and P. Yang. A tabu search heuristic for the local container drayage problem under a new operation mode. *Transportation Research Part E: Logistics and Transportation Review*, 62:136–150, 2014.

[106] R. Zhang, J.-C. Lu, and D. Wang. Container drayage problem with flexible orders and its near real-time solution strategies. *Transportation Research Part E: Logistics and Transportation Review*, 61:235–251, 2014.

[107] R. Zhang, W. Y. Yun, and H. Kopfer. Heuristic-based truck scheduling for inland container transportation. *OR spectrum*, 32(3):787–808, 2010.

[108] R. Zhang, W. Y. Yun, and H. Kopfer. Multi-size container transportation by truck: modeling and optimization. *Flexible Services and Manufacturing Journal*, 27(2-3):403–430, 2015.

[109] R. Zhang, W. Y. Yun, and I. Moon. A reactive tabu search algorithm for the multi-depot container truck transportation problem. *Transportation Research Part E: Logistics and Transportation Review*, 45(6):904–914, 2009.

[110] R. Zhang, W. Y. Yun, and I. K. Moon. Modeling and optimization of a container drayage problem with resource constraints. *International Journal of Production Economics*, 133(1):351–359, 2011.