Routing and Scheduling Problems of Container Trucks in a Shared Resource Environment
Routing and Scheduling Problems of Container Trucks in a Shared Resource Environment

DISSERTATION

submitted in partial satisfaction of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

in Civil Engineering

by

Kyungsoo Jeong

Dissertation Committee:
Professor Stephen G. Ritchie, Chair
Professor R. Jayakrishnan
Professor Will Recker

2017
DEDICATION

To

my wife, my parents and my sister

for their unwavering love and support
# TABLE OF CONTENTS

LIST OF FIGURES .................................................................................................................. v

LIST OF TABLES ..................................................................................................................... vii

ACKNOWLEDGEMENTS ....................................................................................................... ix

CURRICULUM VITAE ........................................................................................................... xi

ABSTRACT OF THE DISSERTATION ................................................................................... xiv

Chapter 1 INTRODUCTION ................................................................................................. 1

1.1 Container truck operation (drayage operation) .......................................................... 5
1.2 Shared transportation for container truck operation .............................................. 7
  1.2.1 Street turning strategy ...................................................................................... 7
  1.2.2 Decoupling strategy ....................................................................................... 8
  1.2.3 Temporal dependency .................................................................................... 10
1.3 Research objectives and general outline of dissertation ...................................... 10

Chapter 2 ROUTING AND SCHEDULING PROBLEM WITH SHARING RESOURCES (BASE MODEL)  14

2.1 Introduction ................................................................................................................. 14
2.2 Literature review ...................................................................................................... 17
2.3 Modeling the problem .............................................................................................. 21
  2.3.1 Problem description ...................................................................................... 22
  2.3.2 Mathematical formulation ............................................................................ 28
2.4 Heuristic algorithm ................................................................................................. 31
  2.4.1 Feasibility ..................................................................................................... 31
  2.4.2 Initial solution .............................................................................................. 38
  2.4.3 Improvements ............................................................................................... 40
2.5 Numerical experiments ........................................................................................... 50
  2.5.1 Test instances .............................................................................................. 50
  2.5.2 Computational results .................................................................................. 54

Chapter 3 SELECTIVE ROUTING AND SCHEDULING PROBLEM WITH SELECTIVE EMPTY CONTAINER PICKUP ................................................................................. 69

3.1 Introduction ................................................................................................................. 69
3.2 Literature review ...................................................................................................... 71
3.3 Modeling the problem .............................................................................................. 76
  3.3.1 Problem description ...................................................................................... 76
3.4. Solution approach ............................................................................................... 83
  3.4.1. Pre-selection algorithm ............................................................................. 83
  3.4.2. Modification to two-stage algorithm ......................................................... 85
3.5 Numerical experiments ........................................................................................... 90
  3.5.1 Test instances .............................................................................................. 90
LIST OF FIGURES

Figure 1-1 Typical drayage processes ........................................................................................................6
Figure 1-2 Comparison of operations with and without street turning strategy .................................8
Figure 1-3 Comparison of operations with and without decoupling strategy ..................................9
Figure 2-1 Potential truck routes with two operational strategies ......................................................25
Figure 2-2 Feasibility Check Algorithm ..................................................................................................32
Figure 2-3 Service start time calculation ...............................................................................................35
Figure 2-4 Service start time improvement ...........................................................................................36
Figure 2-5 Service start time adjustment ...............................................................................................38
Figure 2-6 Initial solution construction algorithm ................................................................................40
Figure 2-7 Neighborhood structures ....................................................................................................44
Figure 2-8 Route addition and reduction processes ............................................................................45
Figure 2-9 VNS Algorithm Flowchart ................................................................................................48
Figure 2-10 GVNS Algorithm Flowchart ............................................................................................50
Figure 2-11 Examples of hypothetical networks by type ....................................................................52
Figure 2-12 Examples of optimal routes and schedules by objective (instance 4L2R) .................60
Figure 2-13 Comparison of % difference of total operation time and total travel time among objectives (box plot) ........................................................................................................................................61
Figure 2-14 Comparison of percentage of performed strategies by objective ...............................62
Figure 2-15 Examples of objective value by customer participation rates in the two operational strategies ........................................................................................................................................64
Figure 2-16 Box plots of objective reductions of feet operations with two strategies against fleet operations without them .............................................................65
Figure 2-17 Comparison of objective reductions of feet operations with two strategies against fleet operations without them by test instance.................................................................66

Figure 3-1 Overall process of the container truck routing problem with collaboration.......79
Figure 3-2. Example of a selective container truck routing problem with collaboration......80
Figure 3-3 Additional neighborhood structures..................................................................................................................87
Figure 3-4 GVNS with random neighborhood structures ...........................................................................................................89
Figure 3-5 Comparison of objective reduction of problem with pre-selection algorithm.....95
Figure 3-6 Comparison of objective by size of pre-selected tasks from outside carriers.....97
Figure 3-7 Comparison of objective according to the exchanging cost............................98
Figure 4-1 Comparison of the proposed problem against a general problem.......................108
Figure 4-2 Additional neighborhood structures......................................................................................................................114
Figure 4-3 Effect of postponements controlled by the weight from instances 10..............123
Figure 4-4 Effect of postponements controlled by the weight from instances 12..............124
LIST OF TABLES

Table 2-1 Travel distance between task nodes ................................................................. 27
Table 2-2 Number of customers by customer set .............................................................. 53
Table 2-3 Parameters for the algorithms ............................................................................ 55
Table 2-4 Comparison of GVNS and VNS against Solver (small size instances) ............... 56
Table 2-5 Comparison of GVNS and VNS against Solver (large size instances) ............... 57
Table 2-6 Summary of performance comparison .............................................................. 58
Table 3-1 Number of customers by customer set including empty pickups from outside ... 90
Table 3-2 Comparison of GVNS against Solver (small size instances) ............................. 92
Table 3-3 Comparison of GVNS against Solver (large size instances) ............................. 93
Table 3-4. Summary of performance ................................................................................. 93
Table 3-5 Comparison of number of empty pickups for collaboration by exchanging cost .. 99
Table 4-1. Comparison of GVNS against Solver for the test instances having all tasks
released on day 1 (small size instances) .......................................................................... 117
Table 4-2 Comparison of GVNS against Solver for the test instances having all tasks
released on day 1 (large size instances) .......................................................................... 118
Table 4-3. Comparison of GVNS against Solver for the test instances having tasks over a
two-day period (small size instances) ............................................................................. 119
Table 4-4 Comparison of GVNS against Solver for the test instances having tasks over a
two-day period (large size instances) ............................................................................. 120
Table 4-5 Summary of performance for the test instances having all tasks released on day 1
............................................................................................................................................... 121
Table 4-6 Summary of performance for the test instances having all tasks released over a two-day period .......................................................... 121

Table A-0-1 Number of customers and trucks for each test instance................................. 135

Table A-0-2. Comparison of GVNS against Solver.......................................................... 136
ACKNOWLEDGMENTS

I would like to thank all teachers and colleagues who have taught me and worked me together. First of all, I would like to acknowledge my academic advisor, Professor Stephen Ritchie. I am very grateful for many research and creative opportunities that he has provided me, and appreciate it. I am particularly thankful to his advising approach and welcoming environment for research areas, which allows me to establish my own motivation and accomplish my research goal during my doctoral study. I also appreciate the enormous instruction and valuable comments of my committee members, Professor R. Jayakrisnan and Professor Will Recker. I would like to express my special thanks to the rest of ITS professors, Professor Wenlong Jin, Professor Michael G. McNally, Professor Jean-Daniel Saphores, and Professor Amelia Regan from whom I have absorbed the content knowledge required of transportation professionals and mature researching skills.

I am truly thankful to my mentors, Dr. Andre Tok, Dr. Joseph Chow and Dr. Jaeyoung Jung. They have been so open to discuss research ideas, and have provided me valuable insight and suggestions on my work. It was a great experience for me to work with them through research projects and papers.

I would like to thank all my friends at ITS, who had valuable discussions and gave me the great idea on my research. Especially, I would like to acknowledge Dr. Kyung Hyun for her personal and academic support. We helped and encouraged each other when we encounter obstacles in our research. I will never forget a lot of memorable movements in my doctoral life with Dr. Suman Mitra, Dr. Ashley Lo, Dr. Daniel Rodriguez, Dr. Iris You and Dr. Jee Eun Kang.
I would like to thank my wife, my parents and my sister who have supported me with unwavering support and love. I would like to acknowledge my wife, Sooyoung, with everlasting gratefulness. She has been my best friend and constant companion, and is only one who has shared the entire journey with me and helped to accomplish it in California far away from our hometown. My doctoral life and this dissertation without her sacrifice, support, and care is unimaginable.

Finally, I am grateful for the financial support and research opportunities provided by the California Department of Transportation and the California Air Resources Board, and the Korean American Society of Civil Engineers.
CURRICULUM VITAE

RESEARCH INTERESTS
Optimization, Vehicle routing problems, Big data analysis, Freight modeling, Intelligent transportation systems, Urban logistics, Advanced traffic management, Data collection and mining

EDUCATION
Ph. D. in Civil and Environmental Engineering (Transportation System Engineering) Jun 2017
University of California, Irvine – Irvine, CA
M.S. in Civil, Urban and Geosystem Engineering (Transportation Engineering) Aug 2006
Seoul National University – Seoul, Korea
Thesis: “A study on Real Time Bus Holding Strategy”, Advisor: Dr. Kyungsoo Chon
B.S. in Civil, Urban and Geosystem Engineering Feb 2004
Seoul National University – Seoul, Korea

PROFESSIONAL EXPERIENCE
Institute of Transportation Studies, University of California, Irvine – Irvine, CA Nov 2011 – Mar 2017
Graduate Student Researcher
• California Truck Data Collection, PI: S.G. Ritchie, Sponsor: California Department Transportation
• California Vehicle Inventory and Use Survey, PI: S.G. Ritchie, Sponsor: California Air Resource Board
• The Requirements and Recommendations for Development of a California Weigh-in-Motion Facility, PI: S.G. Ritchie, Sponsor: California Department Transportation
• California Statewide Freight Forecasting Model, PI: S.G. Ritchie, Sponsor: California Department Transportation
Transportation Research Engineer
• Developed a master plan for implementing new BRTs, LRTs, and express railways in Seoul metropolitan area
• Analyzed transit operations and developed an operational plan for transit system
• Designed urban railroad system and network
• Performed travel demand forecasting, traffic impact studies and traffic volume count projects
Graduate Student Researcher
• Conducted feasibility studies of constructing highways or railways in Korea
• Implemented the four-step model for travel demand forecasting
• Performed benefit-cost analysis to evaluate transportation project

PUBLICATIONS
• Kyungsoo Jeong and Stephen G. Ritchie (2017), Local container truck routing problem with its operational flexibility, proceedings of the Annual Meeting of the Transportation Research Board, Washington, D.C.

• Kyung (Kate) Hyun, Kyungsoo Jeong, and Stephen G. Ritchie (2017), Assessing crash risks considering vehicle interactions with trucks using point detector data, *Journal of Accident Analysis and Prevention, Under review*

• Andre Tok, Kyung (Kate) Hyun, Sarah Hernandez, Kyungsoo Jeong, Yue Sun, Craig Rindt, and Stephen G. Ritchie (2017), Truck activity monitoring system (TAMS) for freight transportation analysis, *Transportation Research Record: Journal of the Transportation Research Board, in press*


• Daniel Rodriguez-Roman, Neda Masoud, Kyungsoo Jeong and Stephen G. Ritchie (2014), A goal-programming approach to allocate Freight Analysis Framework mode flow data. *Transportation Research Record: Journal of the Transportation Research Board 2411, 82-89*


**TECHNICAL SKILLS**

- Programming: Python, Visual Basic, JAVA
- Scientific programming languages: GUROBI OPTIMIZATION, R, MATLAB, AMPL
- Transportation Tools: TransCAD, EMME/3, PARAMICS, TransModeler, TRANSYT-7F, VISSIM, AutoCAD
- Other: PostgreSQL, Quantum GIS, SAS

**CONFERENCE PRESENTATIONS**

• Local container truck routing problem with its operational flexibility

• Truck activity monitoring system (TAMS) for freight transportation analysis

• Influence of personal concerns about travel on travel behavior

• Assessing crash risks considering vehicle interactions with trucks using point detector data

• California Vehicle Inventory and Use Survey: Pilot Study Insights

• Relationship between travel concerns and activity using the 2009 National Household Travel Survey

• A goal-programming approach to allocate Freight Analysis Framework mode flow data

• Estimating Path Flows and Origin-Destination Matrices for Multimodal Freight
HONORS
• Graduate Scholarship, Korean American Society of Civil Engineers 2015
• Travel Grant Award for TRB Conference, Korean Transportation Association in America (KOTAA) 2016-2017
• Merit Based Scholarship from Seoul National University 2004 - 2006

PROFESSIONAL TRAINNIG
• 2015 Transportation Academy, Women Transportation Seminar (WTS) Aug. 2015

PROFESSIONAL SERVICE
• Member: Transportation Research Board, Korean Society of Road Engineers, Korean Transportation Association in America, Korean Society of Transportation
• Reviewer: Transportation Research Record (2015-2016)
• License: Certificate of Engineer Civil Engineering (Korea), Certificate of Engineer Information Processing (Korea)
ABSTRACT OF THE DISSERTATION

Routing and Scheduling Problems of Container Trucks in a Shared Resource Environment

By

Kyungsoo Jeong

Doctor of Philosophy in Civil and Environmental Engineering

University of California, Irvine, 2017

Professor Stephen G. Ritchie, Chair

More frequent vehicle movements are required for moving containers in a local area due to low unit volume that a single vehicle can handle compared with vessels and rails involved in the container supply chain. For this reason, truck operations for moving containers significantly affect not only transportation cost itself but also product price. They have inherent operational inefficiencies associated with empty container movements and container processes at facilities such as warehouses, distribution centers and intermodal terminals. One critical issue facing the trucking industry is the pressing need for truck routing plans that reduce such inefficiencies. Hence, this dissertation proposes to apply the concept of sharing resources, which is an emerging economic model, to container truck operations in order to resolve this issue. Two shareable resources – vehicles and containers – are considered.

This study extends the literature on routing and scheduling problems that arise from container movements, and examines the possible benefits of sharing resources across
customers. A series of truck container routing and scheduling problems were developed by assuming different levels of resource sharing among: (1) customers of one trucking operator, (2) customers across collaborations of multiple operators, and (3) customers over multi-day operations. To enable a trucking company to operate its fleet under a shared resource environment, two operational strategies – street turning and decoupling operations – together with temporal precedence constraints – in addition to the time constraints that are typically included in the vehicle routing problem with time windows (VRPTW) – were adopted to address the proposed problems.

Two meta-heuristic algorithms based on a variable neighborhood search (VNS) scheme were developed to solve the proposed problems, including temporal precedence constraints – which are computationally more expensive – for real-world applications. To address flexible time windows resulting from temporal precedence constraints, a novel feasibility check algorithm was developed.

Results from a series of numerical experiments confirm that the proposed approach leverages the advantages of resource sharing, and the meta-heuristic algorithms are efficient solution approaches for each problem with the targeted resource sharing. Consequently, this dissertation offers a platform for the development of a decision-support tool for drayage companies by applying three different levels of resource sharing into their operations.
CAHPTER 1  INTRODUCTION

Trade in goods is essential to our daily lives as well as to economic activity. In the field of transportation, the main challenge is to improve the efficiency of goods movement. This significantly affects the cost of goods as well as transportation costs. In this context, containerization—which was first introduced in 1956—was a game changer in goods movement, particularly in international trade. Containers are globally standardized and are designed to be transported efficiently and securely over long distances without being opened to be shifted from one mode to another – timely and costly expensive processes in traditional shipments. Since they significantly reduce transshipment costs, time, and complications on the way, not only has the transportation and logistics industry (e.g., intermodal transportation systems) changed, but the shape of the global economy (e.g., market globalization and production globalization) has changed (Levinson, 2006). In recent years, containers have become widely utilized within the domestic share of intermodal movement, as well as in international trade (Association of American Railroads, 2016).

As a result of consistent growth in international and intermodal movement and advancements in technology for shipping and handling containers, the use of cargo containers has increased substantially during recent decades. In the U.S., port container traffic in 2014 was 46 million twenty-foot equivalent units – an increase of 60 percent over 2010 (The World Bank, 2014). Trucks are typically responsible for the first or last segments of container transport, even ones that are transported by vessel or rail in their line-haul operation. According to Caltrans, about 75% of all port-related freight movements are served by truck (Caltrans, 2012). Increased numbers of containers inherently generate
many more truck movements between intermodal terminals and local distribution points. These trucks share congested routes passing through urban areas with general passenger and other commercial vehicles. Consequently, container truck movement has a significant impact on local traffic, economics, and the environment. This is one of the most serious problems facing U.S. and intermodal ports from the point of view of regional transportation planners (Little, 2015). From a supply chain perspective, although the first/last -mile container movements operated by trucks are a relatively small portion of the container supply chain, the cost in moving a container in a local area is relatively high (Zhang, Yun and Moon, 2009; Funke and Kopfer, 2016). This truck operation cost is highly related not only to manufacturers’ profit, but also that of transportation providers. Hence, stakeholders have been trying to find more efficient truck operations for moving containers at various levels (e.g., strategic, operational, long-term, and short-term planning). This can lead to cost savings for an individual business utilizing containers, as well as a reduction in the negative impacts of container trucks on a local community.

This dissertation focuses on an operational problem facing trucking companies in the container transportation industry. A trucking company receives a set of tasks, which entail picking up and delivering containers among intermodal terminals, container yards, and customers’ locations, to and from customers. It assigns trucks to complete all tasks with the given resources and within an operational period, which generates a complex routing and scheduling problem. This problem can be modeled as a routing and scheduling problem of container trucks that are required for pickup and delivery tasks. Such problems are in the class of full truckload pickup and delivery problems (FT-PDP) (Savelsbergh and Sol, 1995;
Zhang, Smilowitz and Erera, 2011). However, a truck directly moves a container between two locations without stopping at intermediate locations, and these two locations are a pickup and delivery location, respectively. In container truck operations, pickup and delivery location for a certain task can be paired, which is defined as a request. If a customer wants to send his container, this needs a movement that a truck picks up the container at the customer location and delivers it, for example, to an intermodal terminal. Because the customer location and intermodal terminal are known in advance, those can be combined into one task node, which can be called a pickup task. Thus, FT-PDP can be reformulated as an asymmetric Traveling Salesman Problem (am-TSP) (Savelsbergh and Sol, 1995; Wang and Regan, 2002; Jula et al., 2005; Smilowitz, 2006), which can help to reduce the problem size and computational complexity. Hence, this dissertation adopts am-TSP formulation and extends it with consideration of temporal constraints.

In the truck operation for moving containers, however, there exists inherent operational inefficiencies caused by empty container movements and container processes at a customer (i.e., packing containers at shippers and unpacking containers at receivers) that container operation itself requires, which are distinct from general routing problems in urban logistics. In order to address these issues, this study proposes routing and scheduling problems in which the concept of shared transportation is incorporated. Under the shared freight transportation environment, two resources—trucks and containers for moving commodities—can be shared among customers and managed by a trucking company separately or by multiple companies together. This study adopts two practical operational strategies that have been proposed to address the inefficiencies: street turning operation
and decoupling operation. Although obvious benefits of these two operational strategies exist, those have not been widely implemented in practice, which is not a straightforward approach. For street turning operation, Islam and Olsen (2014) pointed out the lack of trust between customers, different types of containers and dissimilar geographic locations between shippers and consignees often make this operation infeasible. To overcome the issues, a proper decision tool is required (Deidda et al., 2008), which can provide information such as routing options and container tracking. According to Lai et al. (2013), the carrier’s policy does not allow trucks and containers to be uncoupled during customer service in a daily operation in order to provide high-quality service and prevent possible problems during the service. In addition, decoupling operation strategy in the literature on container truck operations has been usually neglected (Sterzik, Kopfer and Funke, 2015), which increases the complexity of the problem.

Consequently, the proposed problems leverage the advantages of these strategies that enable resource sharing, which could potentially reduce operational inefficiencies. In addition, meta-heuristic algorithms based on a variable neighborhood search (VNS) scheme are developed to solve the proposed problems within a reasonable computational time. Since the problems are an extension of a class of vehicle routing problem (VRP) that is well known to be NP-hard, they are difficult to solve with commercial software, which hinders the real-world application of these problems with larger customers. The proposed approach can allow a decision maker to find solutions for the routing and scheduling problems confronting real-world operations, and can be applied to VRPs having similar constraints to those considered in this study.
1.1 Container truck operation (drayage operation)

Container truck operation, called drayage operation, refers to the regional movement of loaded and empty containers (i.e., containers with a trailer) by truck (i.e., tractor) between customers (i.e., shippers and consignees), intermodal terminals (i.e., ports and rail yards) and equipment yards (i.e., containers and trailer yards) (Smilowitz, 2006). Figure 1-1a and 1-1b represent a typical truck operation for moving containers seen from the perspectives of consignee and shipper, respectively (for more details, see NCFRP Report 11, Tioga Group Inc., 2011). While the main task for a consignee is moving a loaded container from an intermodal terminal to the consignee’s location, that for a shipper is moving a loaded container from the shipper’s location to an intermodal terminal. When a trucking company receives an order for a consignee, it dispatches a truck to the intermodal terminal. After obtaining the loaded container, the truck then delivers it to the consignee. In some cases, the loaded container can be delivered to another intermodal terminal for forwarding to the consignee located far from the terminal (i.e., ongoing movement). After the process of unpacking the container, the empty container should be moved by the trucking company. To meet a shipper’s request, an empty container is delivered to the shipper and then the loaded container is picked up and delivered to the intermodal terminal. The trucking company receives each type of task during an operational period (e.g., a day) and makes a plan of routes and schedules to complete all tasks, utilizing its given resources and minimizing total operating costs. For each customer, a set of trucks repeatedly perform the processes described above. Sometimes, the process can be partial depending on the customer’s request. For example, only empty or loaded container pickups may exist in the truck operation.
In general, truck operation is needed to deliver loads from a supplier to multiple demanders, pick up loads from multiple suppliers to deliver to a demander, or pick up and deliver loads from multiple suppliers to multiple demanders. Each truck trip is related to revenue. As previously mentioned, truck operation for container movement initially arises out of the demand to transfer containers bearing commodities (i.e., loaded containers). In contrast with normal trucking operations, however, a trucking company is required to move containers without commodities (i.e., empty containers) as well, because loaded containers inherently produce empty containers, which generate non-revenue trips. In addition, a customer can be both a supplier and a demander with the successive packing and unpacking of a container in an operation period. As shown in Figure 1-1, for example, a consignee requests an empty container to be moved after receiving a loaded container, and a shipper wants to obtain an empty container to fill with commodities and then send loaded. This container movement requires a relatively long process time for handling containers at
customer locations, which results in a considerable increase in waiting time and a decrease in the utilization of trucks

1.2 Shared transportation for container truck operation

This dissertation incorporates the shared transportation problem into the routing and scheduling problem, which could help to overcome the obstacles facing the traditional container truck operation by focusing on the efficient use of resources that a trucking company has. In container truck operation, there are four resources: drivers, trucks (tractors), container chassis, and containers. Without loss of generality, this study assumes that drivers and chassis can be assigned as required to trucks, and that any location in the system can provide chassis suitable for containers. Thus, containers and trucks are resources that can be shared among customers, which leads to two practical operational strategies: sharing containers refers to a street turning operation strategy, and sharing trucks corresponds to a decoupling operation strategy. These strategies allow a more efficient use of available resources.

1.2.1 Street turning strategy

When customers are willing to share their empty containers with others, street turning strategy can be implemented. Without the strategy, called depot direct, empty containers should be moved back to the container after being picked up at the customer’s location or be supplied from the container yard (see Figure 1-2a). If street turning is allowable, a truck is able to transfer an empty container from its pickup customers to
delivery customers without visiting the container yard (see Figure 1-2b). This approach helps to reduce non-revenue empty trips, saves associated costs and time, and increases the utilization of resources. Despite its considerable benefits, practical limitations and institutional barriers in implementing this strategy have been pointed out (Braekers, Janssens and Caris, 2011). From a modeling perspective, considering street turnings is challenging due to the size of the problems, the existence of time windows, and the need to model multiple resources (Smilowitz, 2006). However, this strategy may have the potential to be put into practice in the increasingly competitive business environment.

![Comparison of operations with and without street turning strategy](image)

**a) Depot direct**  
(no street turning allowed)  
**b) Street turning**

Figure 1-2 Comparison of operations with and without street turning strategy

### 1.2.2 Decoupling strategy

Container truck operation includes the container process time at a customer location. This process time can vary depending on the amount of commodities and the capability of the customer, but is relatively long compared with service time in a general pickup and delivery truck operation. This can generate long dwell time at the customer location when a
truck waits while the container is unpacked or packed, as shown in Figure 1-3a. If a customer allows a truck assigned to his task to be shared with others during his container process, the truck drops off the container at the customer location and then travels to another customer for different tasks, as shown in Figure 1-3b. After a certain process time, the following container can be retrieved by any truck. This approach offers several other benefits, which are a reduction in the truck operation time and the usage of trucks and an increase in the productivity of drivers. Although preventing this strategy results in decreasing flexibility and increasing inefficiency of truck operation, the opportunity to leave a container at a customer location is usually neglected due to the complexity of modeling such an approach (Sterzik, Kopfer and Funke, 2015). When the problem considers decoupling operation strategy with time window, care should be taken to define the time window for the subsequent pickup tasks. The reason for this is that the time window is not determined in advance, as opposed to other general tasks’ time windows.

![Diagram](image1.png)

a) Stay-with (no decoupling allowed)

![Diagram](image2.png)

b) Decoupling

Figure 1-3 Comparison of operations with and without decoupling strategy
1.2.3 Temporal dependency

In order to allow decoupling operations, the proposed problem should consider temporal dependencies between two consecutive tasks, which are loaded container pickup followed by empty container delivery, or empty container pickup followed by loaded container delivery. In addition to the time constraints usually considered in the VRP with time window, temporal dependency is subject to the process time between two consecutive tasks. This means that the earliest available service start time for the following task is determined by the service start time and service time (e.g., container process time in this problem) for the preceding task. These temporal dependencies can be defined as precedence constraints and can be formulated as follows (Dohn, Rasmussen and Larsen, 2011):

\[
\sum_{k \in K} T^k_i + \delta_{ij} \leq \sum_{k \in K} T^k_j, \quad \forall (i,j) \in \Delta,
\]

where \(T^k_i\) represents the service start time when vehicle \(k\) is at customer \(i\), \(\delta_{ij}\) is the process time between \(i\) and \(j\), and \(\Delta\) is a set of customer pairs involved with container processes. This equation is included in the proposed problem, which increases its complexity, but makes it more suitable in practical instances, as will be demonstrated in the following chapters.

1.3 Research objectives and general outline of dissertation

In this dissertation, truck operation plans for moving containers in a local area adjacent to intermodal terminals are studied at an individual truck operator level by
considering the distinctive features of container movement and associated truck movement. The objective is to propose these container truck routing and scheduling problems under a shared transportation environment by extending general mathematical problems that have been applied to drayage operation to incorporate the two operational strategies outlined. The problems consider the total travel distance and operation time together as the objective function, and include additional time constraints that account for temporal dependencies, which make the approach suitable for implementing the two operational strategies. This mathematical approach could provide an optimal truck operation plan to a trucking company by reducing the endogenous inefficiency of container movement. In order to offer a solution approach for the proposed problems, which are well known to be computationally expensive, two meta-heuristic algorithms are proposed and evaluated. As a consequence, this dissertation provides a decision support tool to a decision maker (i.e., truck operation planner) in an individual carrier, which can lead to achieving more efficient systems.

This dissertation is organized as follows. After the Introduction in Chapter 1, three main chapters discuss three different problems, each following the same structure, which is composed of an introduction, a review of relevant literature, modeling the problem, solution approach, numerical experiments, and closing remarks.

In Chapter 2, the container truck routing and scheduling problem for pickup and delivery of loaded and empty containers in a local area is approached by considering two operational strategies: street turning and decoupling operations. In this problem, customers share trucks and containers assigned to them with other customers, and this sharing is
separately controlled by a trucking company. For real-world applications, the problem takes account of site constraints in which each customer can or can not allow each strategy. These site constraints are not applied to the mathematical formulation, but to the routing network to simplify the problem. A two-stage algorithm is proposed as a solution approach. The algorithm consists of an initial solution construction step based on an insertion heuristic and an improvement step based on a VNS algorithm. For the improvement step, two algorithms are developed: a VNS and a general VNS (GVNS) that uses a variable neighborhood descent (VND) as a local search. The performance of algorithms is evaluated and a series of scenarios is analyzed to verify the effects of the two strategies.

Chapter 3 extends the problem proposed in Chapter 2 to one in horizontal collaboration. Resource sharing is applied to customers beyond those that belong to a given trucking company. We assume that empty container pickup tasks can be exchanged with tasks from other companies that participate in the collaboration. In order to support this operation, a well designed web-based system is assumed to exist. In addition, a pre-selection algorithm is proposed that allows a decision maker to select several candidate tasks from a pool of tasks to reduce the size of problem. The GVNS algorithm with the initial solution construction algorithm is utilized to solve the proposed problem, along with several modifications.

A multiday container truck routing and scheduling problem is proposed in Chapter 4. This problem accounts for a limitation that the general problem presented in Chapter 2. The limitation results from the infeasibility generated by a long process time for the consecutive
delivery and pickup tasks. In some cases, a certain task cannot be served within a day, or postponing it to next day can provide a better solution. In order to overcome the limitation and find better solutions, containers and trucks scheduled on a certain day can be shared with ones on other days. Thus, the problem is formulated considering multiday operation with a postponement option. For this problem, the algorithm is modified by adding several neighborhood structures.

Chapter 5 summarizes the proposed schemes and presents the final remarks of this dissertation, indicating further works.

Appendix discusses that the proposed problems can be potentially extended to provide the solutions including the composition of a fleet and routing plan simultaneously in which a fleet is composed of conventional trucks and autonomous trucks.
CHAPTER 2 ROUTING AND SCHEDULING PROBLEM WITH SHARING RESOURCES (BASE MODEL)

This chapter describes a base model of the container truck routing and scheduling problem that occurs in a drayage operation where a set of trucks have to serve pickup and delivery of loaded and empty containers. This model considers sharing two resources, containers and trucks, among customers of a trucking company. For the sharing of resources, the problem incorporates two operational strategies, which are street turning and decoupling operations. In order to simplify the problem, a routing network is modified by combining tasks having fixed origin and destination. The problem is formulated as an asymmetric multiple-vehicle Traveling Salesman Problem with Time Windows (am-TSPTW) including precedence constraints. Meta-heuristic solution approaches are developed based on an insertion heuristic and a variable neighborhood search (VNS) to solve the problem more efficiently. This mathematical model and solution approach will be the basis for the problems presented in the following two Chapters of this dissertation.

2.1 Introduction

Cargo containers involve both domestic and international multi-leg movements and are transported by a combination of transportation modes such as trucks, rail, and ships. Container transportation problems have been addressed to support the container shipping industry and provide optimal solutions to decision makers. As the first and last mile of container movement, containers are frequently transported by truck between intermodal terminals or between customers and intermodal terminals. Since these locations are located
adjacent to or in a metropolitan area (e.g., Los Angeles), container truck movements not only have an impact on local traffic, but are also influenced by this factor.

In a highly competitive market, a trucking company needs efficient planning of truck operations to complete a set of tasks moving containers among customers, intermodal terminals, and container yards, which creates a container truck routing and scheduling problem. However, truck movements for containerized shipment differ from those for non-containerized freight transportation (Bai et al., 2015). Since containers are used as a resource for holding and moving commodities from one location to another, a truck directly transports a container between two locations without stopping at intermediate locations. Although a truck can deal with multiple tasks in one operation, a truck should complete a task (i.e., picking up, moving, and dropping off a container) before undertaking another task. In addition, trucks are requested to move empty containers as well as loaded containers. While customers who want to send their commodities demand empty containers, those who receive their commodities via containers supply empty containers. Consequently, a customer can have two consecutive requests, which entail a pickup of a container followed by a delivery of the same container, for one completed shipment of the container. For example, right after receiving an empty container or loaded container, a customer can request its pickup within a couple of hours when he unpacks or packs the container.

Such characteristics of container truck movements lead to inefficiency in the planning of truck operations as well as in the entire process of container shipment. Since empty containers are usually moved between a container yard and customers’ locations, their
movements generate non-revenue trips, extra time, and costs. Unpacking and packing processes at customers also generate extra time and costs to truck operations. If customers can share a truck and container with others, these inefficient operations might be reduced. To lower the inefficiency of container movement, both academia and industry have proposed two types of operational strategies: street turning and decoupling operations.

Under street turning operations, an empty container can be directly transported between customers if it can be shared between these. While a loaded container task has a fixed origin and destination, either origin or destination of empty containers can be endogenously decided with optimal routes (Zhang, Smilowitz and Erera, 2011). If the street turning strategy is not allowed, an empty container must be moved from a customer to a container yard or from a container yard to a customer. If trucking companies adopt this strategy, they can reduce empty container movements with some degree of coordination between customers and trucking companies (Smilowitz, 2006). The process is that a truck route is constructed by directly connecting a customer supplying an empty container to one demanding an empty container. Attempts have been made to implement this strategy in a real-world application; for example, the Port of Long Beach has been promoting the use of an empty container management system, which helps to share empty containers with other customers (Port of Long Beach, 2008).

Trucks carrying containers are capable of decoupling containers from them, which allows decoupling operations. By adopting this operational strategy, a company can make a decision as to whether a truck stays with the carrying container at a customer location or
leaves for a different customer to serve another task when the customer has two consecutive tasks, such as a delivery task followed by a pickup task. Lai et al. (2013) pointed out that if a truck stays with the carrying container during customer service, customers can have a high quality of service. However, from the operational point of view, the decoupling strategy can reduce the total operating costs, which is beneficial to trucking companies (Xue et al., 2014).

In order to increase the efficiency of container movements and reduce the operation cost of a trucking company, this study extends a full truckload vehicle routing problem by considering both operational strategies simultaneously. In this study, pickups and deliveries of loaded and empty containers are defined as independent tasks from the perspective of customers. The problem also handles the two consecutive tasks separately, which requires a temporal dependency between the two tasks. The problem is formulated as an am-TSPTW, which is a well known computationally expensive problem. Meta-heuristic approaches are proposed to solve the problem efficiently.

2.2 Literature review

Since each container should be directly moved between two locations without midway stops, a container truck routing problem is classified as a full truckload problem. This type of problem can be reformulated as an am-TSP by defining nodes as transportation requests (i.e., loaded trips) (Savelsbergh and Sol, 1995). In this formulation, only distances traveled by trucks without a container between two requests were subjected to the optimal route, since distances traveled with a container were fixed. By considering multiple vehicles and time windows, Wang and Regan (2002) formulated the local truckload pickup and
delivery problem as an am-TSPTW. They applied a window partition-based approach to solving the problem with an iterative method. Jula et al. (2005) also proposed an am-TSPTW to solve a local container problem including social constraints (i.e., drivers’ work-shift hours). They modified the routing network using an approach that calculated the time window for a task node. They developed an exact solution approach, a hybrid genetic algorithm, and an insertion algorithm to solve the problem, and compared the results with different sizes of the problem. Although these two studies included empty container movements, they dealt with such events in the same way as loaded container movements, because they did not consider street turning operation. Also, they did not consider two consecutive tasks. These are major differences from the present study.

Some studies have focused on the fact that each loaded container movement generates an empty container movement before its pickup task and after its delivery task. An empty container coming from a customer who receives a full container can be directly moved to another customer who requests a pickup of a full container when an empty container is available to be shared with others. If a direct route cannot be found to link to such customers requiring empty containers, because of an imbalance between supply and demand of empty containers, empty containers will be moved to a container yard terminal. Under this assumption, Imai et al. (2007) formulated the problem as a variant of a vehicle routing problem with backhaul. To minimize truck operations for the empty container movement, they generated a set of possible pairs of delivery and pickup locations, called merged trips, in advance. In this modified container transportation network, each vehicle departing from a terminal first visits a delivery point and then travels to a pickup point with
an empty container. Hence, their study assumed that a container assigned to a truck should stay with that truck until it returns to the terminal. To find a near optimal solution for these merged trips, they proposed a subgradient heuristic algorithm based on a Lagrangian relaxation by employing a modified heuristic for a bin-packing problem. Caris and Janssens (2009) employed the same concept of merged trips by adding time window constraints into the problem and developing a local search heuristic algorithm with a two-phase insertion heuristic as an initial solution.

By focusing more on empty container movements, drayage truck routing problems have been extended to assign tasks that are based on inbound and outbound container movements at a port or intermodal terminal. This approach led to an empty container repositioning problem. Zhang et al. (2009) categorized customers’ requests for container transportation into four types (i.e., inbound full containers, outbound full containers, inbound empty containers, and outbound empty containers) by direction of container and load type of container from the perspective of the port or intermodal terminal. Each container movement is associated with an empty container; and a set of trucks serves to reposition the empty containers within the routing problem. In this approach, an intermediate path passing through a container yard between two requests was included in the link attributes, which enabled empty containers to be moved. To formulate the minimization of the total unprofitable traveling time caused by empty container trips and unloaded trips, they used an am-TSPTW in which there were multiple container depots and an intermodal terminal. A clustering method and a reactive tabu search algorithm were proposed to solve the problem efficiently. This problem was extended by adding more
intermodal terminals (Zhang, Yun and Kopfer, 2010). Using the same four types of container transport requests, Wang and Yun (2013) formulated the mathematical problem to schedule container operations by two transportation modes, which were trucks and rail. Sterzik and Kopfer (2013) also studied the loaded and empty container routing problem with simultaneous repositioning of empty containers and solved the problem by using an efficient tabu search heuristic. Braekers et al. (2014) proposed a bi-objective approach to solve the container routing problem. Two conflicting objectives—minimizing the number of vehicles and minimizing the total distance traveled—were considered in this problem. They proposed three solution algorithms, which were the iterative method algorithm, the two-phase deterministic annealing algorithm, and the hybrid deterministic annealing and tabu search algorithm, and compared them with each other.

A few studies have considered the detachable characteristic of truck and container and truck operations applying this feature. Smilowitz (2006) assumed that a truck and trailer (container) acted as two resources that could be separated during operations. Under this assumption, they formulated the truck routing problem with flexible tasks as a multi-resource routing problem. However, this study did not include the unpacking and packing time for two consecutive tasks at the same customer. By extending this work, Zhang et al. (2011) developed a dynamic multi-resource routing formulation for a given operating period. They proposed a two-stage stochastic optimization model and applied it to find the optimal routes for given tasks and to provide probabilistic information on dynamic tasks. Xue et al. (2014) proposed the local container drayage problem considering decoupling operations and a tabu search algorithm. They only focused on requests from the empty
pickup nodes and loaded delivery nodes. These two nodes generate the following loaded pickup tasks and empty pickup tasks, respectively. A truck can stay with the container for the following task or travel to serve another task. However, since their study only focused on two types of requests, it did not consider the entire container movement as independent requests and associated empty container repositioning issues. Sterzik et al. (2015) also analyzed the benefit of decoupling operations for drayage truck operation by using a tabu search approach. These two studies considered a time interval for two consecutive tasks, which is close to the approach of this dissertation. However, they used a predefined time window for these tasks, which is different from our study.

 Although a few studies have incorporated shared resources and associated operational strategies in the container truck problem, two operational strategies and flexible time windows for the following tasks, as addressed in this chapter, were not considered simultaneously. Contrary to previous studies, this study proposes a container truck routing and scheduling problem with a shared resource option, in which each pickup and delivery task, including pickup tasks preceded by delivery tasks (i.e., consecutive tasks) is an independent task, and each customer has an option on each operational strategy.

2.3 Modeling the problem

 This study focuses on the container truck routing and scheduling problem from the perspective of individual customers’ requests for pickup and delivery of containers. Two types of containers, loaded containers and empty containers, should be moved by trucks belonging to a trucking company. Container movements among a container yard terminal,
ports, intermodal terminals, and customers (e.g., warehouses and manufacturers) are considered in this problem, which can allow for the distinctive route patterns of drayage trucks identified by You et al. (2016). This problem can be formulated as an am-TSPTW as in previous studies (e.g., Braekers et al., 2014; Wang and Regan, 2002; Zhang et al., 2009).

2.3.1 Problem description

The problem is defined on a general graph $G(N,A)$, where $N$ is the set of nodes and $A$ is the set of arcs. The set of nodes consists of depots $D = \{0, n + 1\}$, an intermodal terminal $T$, a container yard $Y$, and customers $C = \{1,2, ..., n\}$, where $0$ is the departure depot and $n + 1$ is the return depot. According to the task requested by a customer, the set of customers $C$ is categorized into six types: the loaded container delivery customers $C_{fd}$; the loaded container pickup customers $C_{fp}$; the empty container delivery customers $C_{ed}$; the empty container pickup customers $C_{ep}$; the loaded container pickup customers where an empty container is delivered $C_{fped}$; and the empty container pickup customers where a loaded container is delivered $C_{epfd}$. Note that tasks and customers are used interchangeably in this dissertation. The set of arcs $A_C$ defines all pairs of two consecutive tasks $(i, j)$ that require a packing or unpacking process of containers at the same location, where $i \in \{C_{fd}, C_{ed}\}, j \in \{C_{fped}, C_{epfd}\}$. In this problem, the container yard is assumed to have sufficient demand and supply of empty containers to satisfy all the customers’ requests. Each customer has a predetermined service time, and the loaded or empty container delivery customers have a predetermined unpacking or packing time, respectively. While customers in $\{C_{fd}, C_{fp}, C_{ed}, C_{ep}\}$ have a given time window, the time window for customers in $\{C_{fped},$
$C_{epfd}$ is determined based on the service completion time of the preceding task during route construction, which results in time-dependent constraints.

The routing network is modified by considering distinct features of container truck movements and the two operational strategies. In local container movements, while loaded container movements can be designated as well-defined tasks, empty container movements can be expressed as flexible tasks (Zhang, Smilowitz and Erera, 2011). Origin (O) and destination (D) of the loaded container task and corresponding time window are predetermined. Therefore, each OD pair for a loaded container request can be represented by a merged single task node, as shown in Figure 2-1. For these nodes, the time window can be redefined using the time window at O and D and travel time between O and D, based on the method proposed by Jula et al. (2005). In this problem, the time window of the origin task is assumed to be given in advance, and is used to represent that of the merged nodes. Service time $S_i$ for a merged node $i$ is also formulated as the sum of pickup and drop off time at O and D and travel time between O and D. In this network, travel distance on an arc having merged nodes is calculated with reference to O or D of the merged nodes. For example, the distance from a full container pickup task to a full container delivery task is one from D of pickup task to O of delivery task, as shown in Figure 2-1.

Unlike loaded container tasks, an OD pair for empty container requests cannot be merged into a node representing a task. Since this problem allows the street turning strategy, the origin of empty container deliveries and the destination of empty container pickups can be a customer’s location if the tasks select street turning, or a container yard if they take the
depot direct option. Each OD pair for empty containers is determined for the construction of truck routes by considering the locations of tasks and time constraints. Hence, the network should take account of two possible routes: two separate empty container routes through the container yard (see Figure 2-1a) or a route directly connecting an empty container pickup node and a delivery node (see Figure 2-1b).

Decoupling operations also yield a variation of the problem. This problem defines two sets of task pairs \((i, j) \in A_c\) that are affected by decoupling operations: a loaded container delivery \(i \in C_{fd}\) followed by the emptied same container pickup \(j \in C_{epfd}\); and an empty container delivery \(i \in C_{ed}\) followed by the loaded same container pickup \(j \in C_{fped}\). These paired tasks require a packing or unpacking process of containers, which can result in long waiting times at the customer’s location. Thus, a trucking company can run its trucks in the network with two different operational options: staying with the container during the container process (see Figure 2-1c) or decoupling the truck from the container and traveling to serve another task (see Figure 2-1d). A decision on the two different operations has to be made at the route construction level. In the former case, the waiting time for packing or unpacking the container is added to the operation time of the truck, but pickup or drop off time is saved. Since the paired tasks occur at the same location, travel distance and travel time between two consecutive tasks are zero. In the latter case, the following task can be served by the same vehicle or any other vehicle. When the sample vehicle used for the following task under decoupling operations, the vehicle visits other customers before coming back to the location having the remaining task. In this case, not only packing or unpacking time, but also pickup or drop off time should be included in the process time. The
time window for the following tasks $C_{epf_d}$ and $C_{fped}$ is determined when the previous task is completed. In other words, the earliest time window for the following task is calculated as the sum of service completion time of the previous task, the unpacking or packing time of containers at the location, and the pickup or drop off time, where the pickup or drop off time are omitted when a truck stays with the container at the customer. This problem assumes that the end of the time window for the following task is the end of operating hours at the depot.

Figure 2-1 Potential truck routes with two operational strategies
A trucking company serves clients requiring different service qualities and security levels. Thus, each client has a different requirement for sharing containers and trucks in this problem, where site dependencies for each operational strategy exist. Instead of including site constraints in a mathematical formulation, this study deals with the site dependencies at the network level. The network is modified depending on consent to each strategy from each customer. If empty pickup or delivery customers do not want to share their container, which means that they do not allow street turning, a truck is not able to directly travel between these two locations. In this case, the truck must visit a container yard to drop off the container obtained from the empty pickup customer and pick up a new container for the empty delivery customer. On the other hand, if street turning is allowed for these customers, they can be directly connected. If a customer does not adopt decoupling operations, two consecutive tasks \((i, j) \in A_c\) required by the customer cannot be separated. Therefore, these tasks can be combined into one node in a fashion similar to how two nodes for a loaded task are merged. In the modified network, \((C_{fd}, C_{epfa})\) and \((C_{ed}, C_{fped})\) are represented as \(C_{fd}\) and \(C_{ed}\), respectively. To consider container process time between two tasks, packing or unpacking time is added into service time. When these merged nodes are connected to other nodes, they should have the attributes of \(C_{epfa}\) and \(C_{fped}\).

Table 2-1 shows travel distances between two customers (tasks) based on the modified network, where \(d_{ij}\) represents distances traveled between two nodes \(i, j\) and \(d_{iy}\) and \(d_{yj}\) denote distances between a customer and a container yard. When customers are merged nodes and are connected to others, they are designated as O or D of these links. This distance matrix considers site constraints, which means the allowable operations for each
customer as aforementioned. Infeasible arcs can be created when an empty container node is linked directly to another node (Braekers, Caris and Janssens, 2014). Loading or unloading an empty container at the container yard is needed in some cases, for example, when a truck visits an empty delivery customer right after delivering a loaded or empty container. To circumvent this issue, a yard terminal, where a truck can load or unload an empty container, is inserted between the two nodes, and the associated travel distance is added to the arc attributes. Therefore, depending on the combination of customer types, in this routing network there exist two categories of travel distances: \( d_{ij} \) and \( d_{iy} + d_{yj} \).

Using this travel distance matrix and a given speed, travel time between two tasks is calculated. When an arc includes stopping by the container yard, pickup or drop off time of an empty container is added to its travel time. In addition, instead of zero travel time for the same physical location, travel time between two consecutive customers in \( A_C \) is replaced by container process time, which is calculated as container packing or unpacking time minus pickup and drop off time, i.e., \( t_{ij} = P_i - \text{time for pickup or dropoff}, \forall (i,j) \in A_C \). The reason for subtracting pickup and drop off time from the process time is that these customers do not need to pick up or drop off a container at their location.

Table 2-1 Travel distance between task nodes

<table>
<thead>
<tr>
<th>( i \backslash j )</th>
<th>( {D, C_{fd}^{d}}, {C_{fd}^{fd}, C_{fp}, C_{fped}} )</th>
<th>( {C_{ed}} )</th>
<th>( {C_{ed}^{ns}} )</th>
<th>( {C_{ep}, C_{epfd}, C_{f^{nd}}^{nsd}} )</th>
<th>( {C_{ep}, C_{epfd}, C_{f^{nsnd}}^{nsd}} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( {D, C_{fd}^{d}, C_{fp}, C_{fped}} )</td>
<td>( d_{ij} )</td>
<td>( d_{iy} + d_{yj} )</td>
<td>( d_{iy} + d_{yj} )</td>
<td>( d_{ij} )</td>
<td>( d_{ij} )</td>
</tr>
</tbody>
</table>
2.3.2 Mathematical formulation

The problem considers the two proposed operational strategies simultaneously under a shared resource environment. In the application of the model, although minimizing both travel distance and operation time is expected to help to reduce inefficiency of container truck operations, these two factors conflict with each other. Hence, both of these indicators need to be included in the objective. The proposed container truck routing and scheduling problem is thus modeled as follows.

\[ \min Z = \alpha_1 \sum_{v} \sum_{j} x_{i,j}^v + \alpha_2 \sum_{v} \sum_{ij} c_{ij} x_{i,j}^v + \alpha_3 \sum_{v} \left( T_{n+1}^v - T_0^v \right) \]  \hspace{1cm} (2.1)
Subject to

\[ \sum_v \sum_j x_{ij}^v = 1, \forall i \in C \]  
(2.2)

\[ \sum_j x_{ij}^v - \sum_j x_{ji}^v = 0, \forall i \in C, \forall v \in V \]  
(2.3)

\[ \sum_j x_{0j}^v \leq 1, \forall v \in V \]  
(2.4)

\[ \sum_j x_{jn+1}^v - \sum_j x_{0j}^v = 0, \forall v \in V \]  
(2.5)

\[ T_i^v + S_i + t_{ij} - T_j^v \leq M \left( 1 - x_{ij}^v \right), \forall i, j \in N, \forall v \in V \]  
(2.6)

\[ T_i^v + S_i + P_i \left( 1 - x_{ij}^v \right) + t_{ij} x_{ij}^v \leq T_j^v, \forall (i, j) \in A_c, \forall v \in V \]  
(2.7)

\[ a_t \sum_j x_{ij}^v \leq T_i^v, \forall i \in N, \forall v \in V \]  
(2.8)

\[ T_i^v \leq b_t \sum_j x_{ij}^v, \forall i \in N, \forall v \in V \]  
(2.9)

\[ T_{n+1}^v - T_0^v \leq W, \forall v \in V \]  
(2.10)

, where

\( x_{ij}^v \): 1 if vehicle v travels from node i and node j, otherwise 0

\( c_{ij} \): Travel distance from node i and node j

\( t_{ij} \): Travel time from node i and node j

\( V \): Set of vehicles

\( D \): Set of depot, \{0, n + 1\}

\( C \): Set of customers \{1,2, ..., n\}

\( N \): Set of nodes, \( D \cup C \)

\( A_c \): Set of pairs of two consecutive tasks

\( T_i^v \): Service start time at node i by vehicle v

\( S_i \): Duration for service at node i
$P_i$: Duration for packing or unpacking of container at node $i$

$a_i, b_i$: Lower and upper time window at node $i$

$W$: Maximum work-shift hours for driver

$\alpha_1, \alpha_2, \alpha_3$: Weighted factors

$M$: Arbitrary large number

The objective function (2.1) minimizes the weighted sum of the number of trucks used, total travel distance, and total truck operation time. Constraints (2.2) allow every node to be visited exactly once. Constraints (2.3) define flow conservation for each tour. Constraints (2.4) ensure that vehicles can be used for each route at most once. Constraints (2.5) guarantee that each truck leaving from the depot should come back to the depot. Constraints (2.6) indicate that the service start time at each customer is calculated as the service start time at the previous customer, service time at that customer, and travel time between two customers, which ensures sub-tour elimination. Constraints (2.7) define the beginning of the time window for the consecutive tasks $C_{fped}$ and $C_{epfd}$, which is calculated as the sum of service start time of the corresponding $C_{ed}$ or $C_{fd}$, service time, and container process time. If the consecutive task is served with a waiting operation, this saves service time for dropping or picking up a container. On the other hand, if the decoupling occurs at that location, dropping or picking up time should be included to compute the time window, depending on the preceding task. Constraints (2.8) and (2.9) indicate time windows. Constraints (2.10) enforce maximum drivers’ work-shift hours for each tour. Note that the terms “vehicle” and “route” are interchangeable in this problem.
2.4 Heuristic algorithm

An am-TSPTW is well known to be NP-hard (Savelsbergh and Sol, 1995; Jula et al., 2005). To solve this problem efficiently, two-phase heuristic approaches are proposed, based on a variable neighborhood search (VNS) with an initial solution construction algorithm. A VNS is an effective meta-heuristic that systematically explores different neighborhood structures to expand the search area (Mladenović and Hansen, 2001), which has been used in the VRP as well as the m-TSPTW (Da Silva and Urrutia, 2010; Zhao and Chen, 2012). In addition to VNS, a general VNS (GVNS) is also proposed as an improvement method. In extending the VNS, GVNS applies a variable neighborhood descent (VND) to a local search heuristic.

2.4.1 Feasibility

In this heuristic approach, the feasibility of routes should be checked with the time window constraints and the route duration constraints (i.e., constraints 2.8-2.9 and constraints 2.10). As shown in Figure 2-2, the feasibility check algorithm consists of three main steps: service start time calculation, service start time improvement, and service start time adjustment. The algorithm starts with a given set of routes. After the service start time calculation step, time window constraints are checked. If the current service start times for each customer satisfy the corresponding time windows, they will be improved and adjusted through the following steps. Otherwise, the algorithm stops and returns the feasibility index as “infeasible route I.” After carrying out all steps, if all routes satisfy drivers’ work-shift hours, the algorithm returns the final service time with “feasible routes,” otherwise, it
provides the final service time with “infeasible routes II,” which means that the set of routes only violates the route duration constraints. By using the algorithm, the feasibility of routes is assessed and a set of service start times is provided while not only constructing an initial solution, but also improving the solution.

![Feasibility Check Algorithm Diagram](image)

**Figure 2-2 Feasibility Check Algorithm**

**Service start time calculation**

Given a set of routes, travel time matrix, and service time at each customer, the service start time calculation constructs a set of service start times and waiting times, as shown in Figure 2-3. At the same time, the time window for customer \( j \in \{C_{epf_d}, C_{fped} \} \) is updated,
because it is not fixed in advance and depends on the service completion time for preceding customer \( i \in \{ C_{f_d}, C_{ed} \} \), where \( (i,j) \in A_c \). First, we create two lists of customers, \( FC = \{ C_{fped}, C_{epfd} \} \) representing following tasks and \( PC = \{ C_{ed}, C_{fd} \} \) denoting preceding tasks. The calculation is iteratively and sequentially conducted for each customer in each route until all routes are explored. By setting the service start time at the depot \( T^v_0 \) to be the beginning of the time window at the depot \( a_o \), service start time \( T^v_i \) at customer \( i \) is sequentially calculated as \( T^v_i = \max (T^v_{i-1} + S^v_{i-1} + t^v_{i-1,i}, \ a_i) \). This problem allows a truck to wait for the service start time at a customer location. Hence, when the arrival time at customer \( i \) (i.e., \( T^v_{i-1} + S^v_{i-1} + t^v_{i-1,i} \)) is earlier than the earliest time window, the service start time at customer \( i \) is replaced by the earliest time window \( a_i \). In this case, waiting time at customer \( i \) is generated as \( W^v_i = \max (a_i - (T^v_{i-1} + S^v_{i-1} + t^v_{i-1,i}), \ 0) \). However, if customer \( i \) is in the list \( FC \), the calculation stops for the route and moves on to the next route, because the time window for customer \( i \) has not been updated. After updating the service start time for the preceding customer of customer \( i \in FC \), the algorithm can revisit customer \( i \). Thus, when a route visits the following customer \( j \) before the preceding customer \( i \), where \( (i,j) \in A_c \), the route is infeasible.

With \( T^v_i, i \in \{ C_{f_d}, C_{ed} \} \), time window updates for customer \( j \in \{ C_{epfd}, C_{fped} \} \) are implemented in lines 12-19. If the current customer \( i \) is in the list \( PC \), customer \( j \) is identified, and its time window is updated depending on operational status. When customer \( i \) is followed by customer \( j \) in the same route, which indicates that the truck stays with the container, pickup or drop off time is not required, and thus the earliest time window for
customer $j$ is computed as $a_j = T_i^v + S_i + t_{ij}$. On the other hand, in the case of a decoupling operation, pickup or drop off time is added into the equation (i.e., $a_j = T_i^v + S_i + P_i$).

If each customer in the routes has not violated their time window and the sequence of visits, the set of routes is provisionally feasible. Then, the algorithm moves on to the next step with the current service start times, waiting times, and time windows.

1: Set $FC = \{C_{fped}, C_{epfd}\}, PC = \{C_{ed}, C_{fd}\}$

2: While all routes are examined:

3: for route $v$ in a set of routes:

4: Set $T_0^v = a_o$

5: for customer $i$ in route $v$:

6: if $i \in FC$:

7: Break

8: else:

9: $T_i^v = \max (T_{i-1}^v + S_{i-1}^v + t_{i-1,i}, a_i)$

10: $W_i^v = \max (a_i - (T_{i-1}^v + S_{i-1}^v + t_{i-1,i}), 0)$

11: end if

12: if $i \in PC$:

13: Set $j \in FC$ corresponding to $i \in PC$ and remove $j$ from $FC$

14: if $j = i+1$:

15: $a_j = T_i^v + S_i + t_{ij}$

16: else:

17: $a_j = T_i^v + S_i + P_i$

18: end if

19: end if

20: end for

21: end while
**Service start time improvement**

This problem includes minimizing the term representing the total operation time of vehicles in the objective with route duration constraints. Because in the previous step service start times are determined as early as possible, some customers have a waiting time. In order to minimize route operation time, a service start time improvement step is proposed to decrease waiting times of routes as much as possible, as shown in Figure 2-4. Forward time slack, originally proposed by Savelsbergh (1991), is a method that moves service start time for a customer forward while maintaining the feasibility of routes. Markov et al. (2016) applied this concept to examine customers backwards in a route without increasing the complexity of updating service start times. By traversing a route backward, if customer $i$ has waiting time, service start time for the previous customer $i-1$ is shifted forward as far as possible in lines 5-7. Then, the reduction of slack is added to the waiting time for customer $i-1$, but is subtracted from the waiting time for customer $i$ in lines 8-9. During the iteration, the time window for customers belonging to $FC$ is updated in the same manner as in the previous step in lines 10-17.

Generally, this approach in VRPTW ensures the feasibility of routes. However, precedence constraints are incorporated in this problem, and associated time window updates are embedded in this step. Consequently, in some cases, the updated service start time obtained through the forward time slack method cannot respect the updated time window. An additional step is required after service start time improvement if the current status is infeasible.
1: $T, W$, and updated time windows obtained from service start time calculation
2: While all routes are examined:
3: for route $v$ in a set of routes:
4: for customer $i$ in reverse order in route $v$:
5: if $W_i^v > 0$:
6: $T_{i-1}^v = T_{i-1}^v$
7: $T_i^v = \min (T_{i-1}^v + W_i^v, b_{i-1})$
8: $W_{i-1}^v = W_{i-1}^v + (T_{i-1}^v - T_{i-1}^v)$
9: $W_i^v = W_i^v - (T_{i-1}^v - T_{i-1}^v)$
10: if $i - 1 \in PC$:
11: Set $j \in FC$ corresponding to $i - 1 \in PC$
12: if $j = i - 1$
13: $a_j = T_{i-1}^v + S_{i-1} + t_{i-1,j}$
14: else:
15: $a_j = T_{i-1}^v + S_{i-1} + P_{i-1}$
16: end if
17: end if
18: end if
19: end for
20: end while

Figure 2-4 Service start time improvement

**Service start time adjustment**

To resolve the issue addressed in the previous section, service start time adjustment is proposed as shown in Figure 2-5. Because of the updated time window for customer $i \in FC$, service start time $T_i^v$ obtained from the improvement step can be scheduled before the earliest arrival time $a_i$ or after the latest arrival time $b_i$. The proposed adjustment step
systematically tries to shift service start times forward or backward until they are within their time window.

If any customer \( i \in FC \) who violates a time window exists, let \( A = \max (a_i - T_i^v, T_i^v - b_i, 0) \), \( \forall i \in FC, \forall v \in V \), and let \( v' \) and \( i' \) be the corresponding route and customer, respectively. The reason for selecting the customer having the maximum violated time is that this can reduce the number of iterations for the adjustment. If there is slack time between service time and the earliest or latest arrival time, at first, all service start times in route \( v' \) can be shifted by using this slack time in lines 6-12. If service start time \( T_i^{v'} \) for customer \( i' \) in route \( v' \) is earlier than the earliest arrival time \( a_i' \), slack time \( A' \) can be expressed as \( \min (b_i - T_i^{v'}, A) \), \( \forall i \) in route \( v \), and all service start times in route \( v' \) are moved forward by \( A' \). On the other hand, if \( T_i^{v'} \) is later than the latest arrival time \( b_i' \), slack time \( A' \) can be expressed as \( -\min (T_i^{v'} - a_i, A) \), \( \forall i \) in route \( v' \), and all service start times in route \( v' \) are relocated backward by \( -A' \). Through this procedure, service start time can be adjusted without increasing total travel time, violating the time windows for other customers, or revisiting the preceding customer of customer \( i' \).

If slack time does not exist in route \( v' \), the preceding customer \( i'' \) of customer \( i' \) and its route \( v'' \) are identified, and then its service time should be adjusted. Doing so, the earliest arrival time \( a_i' \) for customer \( i' \) can also be adjusted in lines 14-17. This procedure requires shifting \( T_i^{v''} \) for all preceding customers \( i \in \{0, \ldots, i''\} \) forward in time. At each iteration, time window updates for customers belonging to \( FC \) are implemented in the same fashion as in the previous steps.
Set $FC = \{C_{fpd}, C_{epf}, C_{ed}, C_{fd}\}$, $PC = \{C_{ed}, C_{fd}\}$

1. Set $FC = \{C_{fpd}, C_{epf}, C_{ed}, C_{fd}\}$, $PC = \{C_{ed}, C_{fd}\}$
2. $T, W$, and updated time windows obtained from service start time improvement
3. While $\max (a_i - T_i^v, T_i^v - b_i, 0) > 0$, $\forall i \in FC, \forall v \in V$
4. $A = \max (a_i - T_i^v, T_i^v - b_i, 0)$, $\forall i \in FC, \forall v \in V$
5. $(v', i') = \arg \max (a_i - T_i^v, T_i^v - b_i, 0)$, $\forall i \in FC, \forall v \in V$
6. if $a_{i_i} - T_{i_i}^v > 0$:
7. $A' = \min (b_i - T_i^v, A)$, $\forall i$ in route $v'$
8. else if $T_{i_i}^v - b_i > 0$:
9. $A' = -\min (T_i^v - a_i, A)$, $\forall i$ in route $v'$
10. if $|A'| > 0$:
11. $T_i^v'' = T_i^v + A'$, $\forall i$ in route $v'$
12. Update time window
13. else:
14. Set $i'' \in PC$ corresponding to $i' \in FC$
15. Set $v''$ where $i''$ exists
16. $T_i^{v'''} = T_i^{v''} - A$, $\forall i \in \{0, \ldots, i''\}$ in route $v''$
17. Update time window
18. end if
19. end for
20. end while

Figure 2-5 Service start time adjustment

2.4.2 Initial solution

A feasible initial solution is generated based on an insertion heuristic (Solomon, 1987; Ioannou, Kritikos and Prastacos, 2001; Markov, Varone and Bierlaire, 2016). Figure 2-6 represents a modified insertion heuristic for the proposed problem. First, a list of customers in $C_{fd}, C_{fp}, C_{ed},$ and $C_{ep}$ is set as unassigned customers UC with an empty route. The
remaining customers who are in $C_{f_{ped}}$ and $C_{ep_{fd}}$ are added to the list during the route construction procedure, because these customers do not have to be visited before the preceding task is completed, whereupon their time window is determined. Until the list of unassigned customers is empty, the insertion heuristic is applied to search for the best customer and its best insertion location with the lowest cost in the current route.

At each iteration, for all possible locations in the current route, each customer in $UC$ is inserted into a location, and the cost and feasibility of the route is then evaluated in lines 4-13. While checking the feasibility of the routes proposed in the previous section, the algorithm updates service start times and time windows (if needed) for the current route constructed so far. The current route and the set of unassigned customers are updated to include and exclude the selected best customer, respectively. In the meantime, if the inserted customer has the following task, which means that it is in $C_{fd}$ and $C_{ed}$ and generates the following task $C_{ep_{fd}}$ or $C_{f_{ped}}$, the following task is added to the set of unassigned customers in lines 15-18. If no more customers can be inserted into the route due to the infeasibility of the routes, the algorithm creates a new empty route and then repeats the best insertion with the updated $UC$ list until all customers are assigned to the routes.
1: Set a list of unassigned customers $UC = \{C_{fd}, C_{fp}, C_{ed}, C_{ep}\}$ and create a route $R_k$
2: While $UC = \emptyset$:
3: \hspace{1cm} $Cost = +\infty$
4: \hspace{1cm} for $(i, j)$ in a route:
5: \hspace{2cm} for $m$ in $UC$:
6: \hspace{3cm} $R_k(i, m, j) \leftarrow$ insert $m$ between customer $i$ and $j$
7: \hspace{2cm} $Cost_{in} \leftarrow Z(x)$
8: \hspace{2cm} if $R_k(i, m, j)$ is feasible and $Cost_{in} < Cost$:
9: \hspace{3cm} $Cost \leftarrow Cost_{in}$
10: \hspace{3cm} $R_k \leftarrow insert \ m \ with \ (i, m, j)$ and $m_{in} \leftarrow m$
11: \hspace{end if}
12: \hspace{end for}
13: \hspace{end for}
14: \hspace{if $m_{in}$ is founded:}
15: \hspace{Update $R_k$ and Exclude $m_{in}$ from $UC$}
16: \hspace{If $m_{in} \in \{C_{fd}, C_{ed}\}$:}
17: \hspace{Update $UC$ with a customer $j \in \{C_{fpd}, C_{fpd}\}$, $(m_{in}, j) \in A_c}$
18: \hspace{end if}
19: \hspace{else:}
20: \hspace{$k += 1$ and create a new route}
21: \hspace{end if}
22: \hspace{end while}

Figure 2-6 Initial solution construction algorithm

2.4.3 Improvements

A modified VNS and GVNS are proposed to improve the current solution and find the near optimal solution based on a VNS scheme. A VNS, originally proposed by Mladenović and Hansen (2001), iteratively uses a perturbation step to escape from a local optimum and
a local search step to find a local optimum solution. The key idea of the VNS scheme is that local optimal solutions obtained from two different neighborhood structures cannot be identical (Salehipour et al., 2011). Therefore, by applying different neighborhood structures during iterations, this approach can increase the capability to explore wider search spaces and avoid being stuck in a local optimal solution. This meta-heuristic approach has been presented to solve mixed integer problems successively and efficiently, and implemented for various type of VRPs. In particular, a VNS and its variants have been applied to solve TSPs (Mladenović and Hansen, 2001; Da Silva and Urrutia, 2010; Salehipour et al., 2011; Zhao and Chen, 2012), VRPs (Polacek, Hartl and Doerner, 2004; Lei, Laporte and Guo, 2012; Xu, Wang and Yang, 2012; Markov, Varone and Bierlaire, 2016), and location or inventory routing problems (Mjirda et al., 2012; Popović, Vidović and Radivojević, 2012; Escobar et al., 2014).

In a VNS approach, a predetermined local search method including one or two neighborhood structures is typically used, which can sometimes yield difficulties in escaping from a local optimal solution (Escobar et al., 2014). For solving large problems or problems having complex constraints, applying various neighborhood structures is a way to solve the problem efficiently. Therefore, the VNS algorithm proposed in this study tries to use different neighborhood structures in a local search. The proposed GVNS utilizes a VND as a local search, this being a variant of VNS that excludes a perturbation step of VNS and deterministically searches for the solution by changing neighborhood structures with a specific order. On the other hand, the proposed VNS uses one neighborhood structure for a local search, but systematically changes it at each iteration. In other words, a neighborhood structure used in the perturbation step and one used in the local search step are changed at
every iteration. Thus, many different combinations of neighborhood structures can be applied.

**Neighborhood structures**

A set of neighborhood structure is a crucial component of the proposed algorithm. By sequentially changing not only a set of neighborhood structures $N_k$ at the perturbation stage, but also a set of neighborhood structures $N_i$ at the local search stage, the algorithm will search for the improved solution until it meets the stopping criteria. For the proposed problem, eight neighborhood structures that are commonly used in heuristic approaches for the VRP are considered. Figure 2-7 represents how each structure works to generate neighborhoods of current routes. In this figure, blue and green customers represent randomly selected locations for change, dashed lines denote disconnected lines during the operation, and black lines are new sequential connections between customers after an operation. Given the structures, some of them will be selected for a set of neighborhood structures $N_k$ and a set of neighborhood structures $N_i$, respectively. The neighborhood structures used in the algorithm are described as follows.

- Intra-route insertion $N_1$ (Figure 2-7a): A customer is randomly selected from its current location of a randomly selected route and inserted into another location of the same route.
- Inter-route insertion $N_2$ (Figure 2-7b): A customer is randomly selected from its current location of a randomly selected route and inserted to a location of a different randomly selected route.
• Intra-route double insertion $N_3$ (Figure 2-7c): Two consecutive customers are randomly selected from their current location of the route and reinserted to another location of the same route.

• Inter-route double insertion $N_4$ (Figure 2-7d): Two consecutive customers are randomly selected from their current location of the route and reinserted to another location of a different randomly selected route.

• Intra-route 2-opt $N_5$ (Figure 2-7e): Two customers are randomly selected from the same route and their positions are switched with each other.

• Inter-route 2-opt $N_6$ (Figure 2-7f): Two customers are randomly selected from two different routes and their positions are switched with each other.

• Intra-route swap $N_7$ (Figure 2-7g): Two arcs are selected and removed from the same route. Two preceding customers of each arc and two following customers of each arc are reconnected, thus the order of the two arcs is reversed.

• Inter-route swap $N_8$ (Figure 2-7h): Two arcs are selected and removed from two different routes and their endpoints are swapped with each other, which yields the routes of customers behind these points being changed.

![Diagrams](image)

a) Intra-route insertion  
b) Inter-route insertion
This problem assumes that a trucking company has a sufficient number of vehicles in its fleet and can use as many as it needs to complete all the requests, without having a fixed number of vehicles for operations. Instead, to deduce the operation cost for using additional vehicles, the objective includes the cost term of the number of vehicles operated in the solution. Therefore, route addition and reduction procedures should be considered in the improvement procedure. While applying inter-route insertion and inter-route double
insertion, the neighborhood structure creates an empty route in the current set of routes, which can result in adding a route to the neighborhoods, as shown in Figure 2-8a and Figure 2-8c. In addition, those two procedures may remove one route having one customer (Figure 2-8b) or two customers (Figure 2-8d) from the current route set.

Variable Neighborhood Search (VNS)

The proposed VNS approach follows a typical VNS scheme, which consists of perturbation and local search. Figure 2-9 represents the flowchart of the VNS. Before the algorithm is run, neighborhood structures $N_k$ at perturbation level, $N_l$ at local search level,
and stopping criteria should be defined in advance. There are two types of stopping criteria: maximum number of iterations \((\text{maxIter})\) and maximum number of non-improvements \((\text{maxNImp})\). The improvement of the solution starts with initialized parameters (i.e., \(k, l, iCnt, \) and \(nICnt\), where \(iCnt\) is a variable of iteration counts and \(nICnt\) is a variable of non-improvement counts for the solution) and a feasible initial solution obtained from the insertion heuristic. However, when a problem is very constrained, like our case, the insertion heuristic may not be able to provide a feasible solution, because a few customers may not find the best insertion location (Xiao and Konak, 2016). In this case, the perturbation step can easily create a feasible solution to replace the current infeasible initial solution.

At the perturbation step, a selected neighborhood structure creates a set of neighborhoods with a given sample size. In our case, a sample size of five is used. Among these, the neighborhood \(x'\) having the lowest cost is selected as a temporary solution. Perturbation allows the solution to be infeasible with respect to route duration constraints, but not to time window constraints. In other words, solutions labeled “infeasible routes II” as well as “feasible routes” from the feasibility check algorithm are permissible candidates for the perturbation solution. To enable the use of this type of infeasible solution, a penalized objective function has commonly been used in heuristic approaches (Xue et al., 2014; Markov, Varone and Bierlaire, 2016). In a way similar to that proposed by Markov et al. (2016), this study adopts a penalized factor to calculate the cost, expressed as \(F(x) = Z(x) + \inf F \times C_p(x)\), where \(Z(x)\) represent the original objective cost, \(C_p(x)\) is the sum of time for each route over the duration, and \(\inf F\) is a weight factor that is systemically updated during
iterations. In this problem, the penalty factor $inf F$ is initially set to be 1.05. If a solution is infeasible, the penalty factor $inf F$ is increased by 0.05, otherwise it is decreased by 0.02 at each iteration. However, this value cannot be less than 1 or greater than 2.

Using the temporary solution $x'$ obtained from the perturbation, the local search then finds the best local solution $x''$. In the local search, one of the neighborhood structures is randomly selected and changed during the master level iteration. Compared with the perturbation, the local search only examines the solution labeled “feasible routes.” After the local search, the solution $x''$ is evaluated with the current best solution $x^b$. If the local optimal solution is better than the best solution, the best solution is replaced by the local solution, the non-improvement count ($nICnt$) is set to be one, and the algorithm runs with goes to the first neighborhood structure in the predefined set. Otherwise, $nICnt$ is increased by one and the next neighborhood structure is used to search a solution. In the meantime, a neighborhood structure for local search is randomly selected. This procedure continues until no improvement is found using the final structure of the neighborhood set.

Finally, iteration count $iCnt$ is updated. If the algorithm does not satisfy two stopping criteria (i.e., the maximum number of iterations and the maximum number of non-improvements), it goes back to the perturbation step with resetting the set of neighborhood structures $N_k$ and $k$. To change the order of neighborhood structures, the current structures are randomly shuffled. Since different sets of neighborhood structures can be determined for perturbation and local search, different combinations of neighborhood structures can be implemented at each iteration, which is a variant of a general scheme of VNS. For example,
while inter-route swap and intra-route insertion can be used at an iteration for perturbation and local search, respectively, inter-route insertion and inter-route double insertion can be applied at another iteration. If the algorithm reaches the stopping criteria, it is terminated and returns the best solution.

![VNS Algorithm Flowchart](image)

**Figure 2-9 VNS Algorithm Flowchart**
**General Variable Neighborhood Search (GVNS)**

The proposed GVNS employs a basic VNS scheme in the same manner as in the previous section. However, there are three main differences between the two algorithms. First, GVNS adopts a VND approach as the local search. A VND is a variant of VNS that has multiple neighborhood structures and deterministically changes them to find an optimal local solution. As shown in Figure 2-10, given a solution, VND finds the best solution using a neighborhood operator. If the solution is improved, the algorithm searches the solution again with the first operator. Otherwise, it explores the solution with the next operator. This process is continued until there is no improvement. Since this local search can explore wider areas than a local search method having one or two neighborhoods, this local approach has an advantage when a problem is complex (Mladenović and Hansen, 2001).

Second, in the perturbation step of the GVNS, a ban list is introduced to prevent creating the same solution obtained from the previous iterations. The proposed VND algorithm is designed in a deterministic way so that it finds the same solution with the same input solution. In order to avoid having the same input solution for the local search, perturbation selects a temporary solution that is not in the ban list.

Lastly, the adoption of a solution acceptance rule is proposed in the algorithm after the local search. By using this approach, the algorithm can accept not only an improved solution, but also a worse solution with probability $e^{(F(x^b) - F(x^r))/T}$, which is based on a technique used in the simulated annealing (SA) heuristic proposed by Kirkpatrick et al.
This probabilistic acceptance rule allows the GVNS to escape from a local optimum more quickly (Zhao and Chen, 2012).

Figure 2-10 GVNS Algorithm Flowchart

2.5 Numerical experiments

2.5.1 Test instances

In order to test the proposed mathematical formulation and meta-heuristic algorithm, a set of instances was randomly generated based on a Euclidean plane of size 50 miles by 50
miles. Four types of instances are considered according to the location of an empty container yard terminal and the distribution of customers, as shown in Figure 2-11. Location set 1 represents a depot located in the middle of the area, while the container yard is located at the lower left corner of the area, close to the intermodal terminal. Location set 2 depicts a container yard that is located close to the depot. As regards the customer distribution, all customers are uniformly distributed over the area. In another type, senders (e.g., exporters or shippers), which are in \(\{C_{fp}, C_{ed}\}\), and receivers (e.g., importers or consignees), which are in \(\{C_{fd}, C_{ep}\}\), are segregated and clustered within different respective halves of the area.

First, customer locations are randomly created by the type of location depending on the number of customers in \(\{C_{fd}, C_{fp}, C_{ed}, C_{ep}\}\). Those customers having the consecutive task (i.e., \(C_{ed}\) and \(C_{fd}\)) can generate the following task (i.e., \(C_{fped}\) and \(C_{epfd}\)). Since two consecutive tasks occur at the same location, the location of the customers in \(C_{fped}\) and \(C_{epfd}\) is generated corresponding to the location of the customers in \(C_{ed}\) and \(C_{fd}\). In these test instances, it is assumed that all customers in \(C_{ed}\) or \(C_{fd}\) have the following request.
Time windows for each customer, depot, and container yard are randomly generated as follows:

- Time window for the depot and container yard: 6 am - 12 am
- The beginning of time window for customer $C_{fd}, C_{ed}$: a uniform random variable from 8 am to 12 pm, which ensures that the following task can be completed during the operation
• The beginning of time window for \( C_{fp}, C_{ep} \): a uniform random variable from 8 am to 6 pm

• The time window interval for customer \( C_{fd}, C_{fp}, C_{ed}, C_{ep} \): a uniform random variable from 60 minutes to 240 minutes, in 30-minute increments

• The end of time window for \( C_{fpd}, C_{epfd} : 12 \) am.

Pickup and drop off times for containers are assumed to be 10 minutes, and the packing and unpacking times of containers are set to be a uniform random value from 60 minutes to 120 minutes, in 30-minute increments. All distances are Euclidean and the travel times are calculated using a 35 mph truck speed. Drivers’ work-shift hours are assumed to be 10 hours. Fifteen customer sets are generated with different combinations of requests, as shown in Table 2-2. Consequently, a total of 60 test instances is created with four types of instance sets.

<table>
<thead>
<tr>
<th>Customer set</th>
<th>( C_{fd} )</th>
<th>( C_{fp} )</th>
<th>( C_{ed} )</th>
<th>( C_{ep} )</th>
<th>( C_{epfd} )</th>
<th>( C_{fpd} )</th>
<th>total</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>6</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>8</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>10</td>
</tr>
<tr>
<td>6</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>12</td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>14</td>
</tr>
<tr>
<td>8</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>4</td>
<td>3</td>
<td>16</td>
</tr>
<tr>
<td>9</td>
<td>5</td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>5</td>
<td>4</td>
<td>20</td>
</tr>
<tr>
<td>10</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>25</td>
</tr>
<tr>
<td>11</td>
<td>7</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>7</td>
<td>4</td>
<td>30</td>
</tr>
<tr>
<td>12</td>
<td>5</td>
<td>7</td>
<td>10</td>
<td>3</td>
<td>5</td>
<td>10</td>
<td>40</td>
</tr>
</tbody>
</table>
2.5.2 Computational results

Evaluation of performance

The mathematical problem and the proposed solution algorithms were coded in Python and run on a Mac PC with a 2.5 GHz i5 processor and 16 GB of RAM. Gurobi (ver. 6.0.1), which is a commercial solver for mixed integer problems, was used to solve the problem with the given mathematical formulation. Solutions obtained from the solver were compared with results solved by the proposed heuristic algorithms to validate them and assess their performance. In general, the computation time of a solver increases significantly in finding the optimal solution in vehicle routing and scheduling problems as the size of problem increases. Thus a CPU time limit of 3600 seconds was forced on the solver. To evaluate the performance of the proposed heuristic algorithms, the best objective value, average value of objective values, and average run time were obtained after ten runs. Table 2-3 represents a set of neighborhood structures for master level and local search and the two stopping criteria that were used to run each algorithm. While two sets of neighborhood structures for VNS (i.e., $N_k$ and $N_l$) were designed to change at every iteration, those for GVNS had a deterministic order, which is obtained from the preliminary experiments. In GVNS, the initial temperature parameter $T$ for the acceptance threshold adopted by a SA was set to be 100 and was decreased by 10% at every iteration.
In this experiment, we assume that a trucking company wants to prioritize minimizing the use of trucks over travel distance and operation time. Average truck speed is used as a conversion factor to value operation time, which results in time being an equivalent unit to travel distance. Thus, the weight factors of the objective are assumed to be $\alpha_1 = 100$, $\alpha_2 = 1$, $\alpha_3 = 35/60$.

Assuming that all customers allow the two operational strategies, the test instances generated in this experiment were solved by the commercial solver, GVNS, and VNS with the corresponding parameters. Table 2-4 and Table 2-5 represent the results of small and large size instances, respectively. Even in small size instances, the solver could not find the optimal solution within the given time limit from instances having over 12 customers. The reason for this is that the problem has an objective including both total operation time and temporal constraints, which was also observed from a similar study conducted by Funke and Kopfer (2016). Their result showed that a commercial solver took much shorter to minimize the total travel distance than to minimize the total operation time. While the solver was able to find the optimal or current incumbent solution representing with gap for small size
instances, for test instances having over 40 customers it could not find a solution within the given time limit.

Table 2-4 Comparison of GVNS and VNS against Solver (small size instances)

<table>
<thead>
<tr>
<th>Instances</th>
<th>Solver(Gurobi)</th>
<th>GVNS</th>
<th>VNS</th>
</tr>
</thead>
<tbody>
<tr>
<td>01L1R</td>
<td>515.24</td>
<td>0.10</td>
<td>515.24</td>
</tr>
<tr>
<td>01L1C</td>
<td>522.83</td>
<td>0.28</td>
<td>522.83</td>
</tr>
<tr>
<td>01L2R</td>
<td>515.24</td>
<td>0.10</td>
<td>515.24</td>
</tr>
<tr>
<td>01L2C</td>
<td>522.83</td>
<td>0.13</td>
<td>522.83</td>
</tr>
<tr>
<td>02L1R</td>
<td>745.62</td>
<td>0.12</td>
<td>745.62</td>
</tr>
<tr>
<td>02L1C</td>
<td>796.23</td>
<td>0.16</td>
<td>796.23</td>
</tr>
<tr>
<td>02L2R</td>
<td>731.48</td>
<td>0.17</td>
<td>731.48</td>
</tr>
<tr>
<td>02L2C</td>
<td>774.15</td>
<td>0.17</td>
<td>774.15</td>
</tr>
<tr>
<td>03L1R</td>
<td>745.94</td>
<td>0.43</td>
<td>745.94</td>
</tr>
<tr>
<td>03L1C</td>
<td>776.50</td>
<td>0.44</td>
<td>776.50</td>
</tr>
<tr>
<td>03L2R</td>
<td>773.49</td>
<td>0.54</td>
<td>773.49</td>
</tr>
<tr>
<td>03L2C</td>
<td>763.63</td>
<td>0.42</td>
<td>763.63</td>
</tr>
<tr>
<td>04L1R</td>
<td>935.83</td>
<td>2.50</td>
<td>935.83</td>
</tr>
<tr>
<td>04L1C</td>
<td>1033.92</td>
<td>8.06</td>
<td>1033.92</td>
</tr>
<tr>
<td>04L2R</td>
<td>773.60</td>
<td>10.71</td>
<td>773.60</td>
</tr>
<tr>
<td>04L2C</td>
<td>830.06</td>
<td>20.09</td>
<td>830.06</td>
</tr>
<tr>
<td>05L1R</td>
<td>1283.07</td>
<td>799.13</td>
<td>1283.07</td>
</tr>
<tr>
<td>05L1C</td>
<td>1837.21</td>
<td>934.01</td>
<td>1837.21</td>
</tr>
<tr>
<td>05L2R</td>
<td>1113.46</td>
<td>824.97</td>
<td>1113.46</td>
</tr>
<tr>
<td>05L2C</td>
<td>1279.33</td>
<td>449.36</td>
<td>1279.33</td>
</tr>
<tr>
<td>06L1R</td>
<td>1514.38</td>
<td>3600.00</td>
<td>991%</td>
</tr>
<tr>
<td>06L1C</td>
<td>1810.09</td>
<td>3600.01</td>
<td>26.5%</td>
</tr>
<tr>
<td>06L2R</td>
<td>1115.20</td>
<td>3600.00</td>
<td>23.36%</td>
</tr>
<tr>
<td>06L2C</td>
<td>1219.37</td>
<td>3600.01</td>
<td>27.45%</td>
</tr>
<tr>
<td>07L1R</td>
<td>1410.35</td>
<td>3600.01</td>
<td>33.80%</td>
</tr>
<tr>
<td>07L1C</td>
<td>1668.98</td>
<td>3600.01</td>
<td>34.29%</td>
</tr>
<tr>
<td>07L2R</td>
<td>1276.15</td>
<td>3600.01</td>
<td>33.11%</td>
</tr>
<tr>
<td>07L2C</td>
<td>1478.90</td>
<td>3600.01</td>
<td>37.35%</td>
</tr>
<tr>
<td>08L1R</td>
<td>1926.49</td>
<td>3600.00</td>
<td>46.73%</td>
</tr>
<tr>
<td>08L1C</td>
<td>1955.30</td>
<td>3600.01</td>
<td>42.58%</td>
</tr>
<tr>
<td>08L2R</td>
<td>1499.73</td>
<td>3600.01</td>
<td>45.28%</td>
</tr>
<tr>
<td>08L2C</td>
<td>1587.66</td>
<td>3600.01</td>
<td>43.54%</td>
</tr>
<tr>
<td>09L1R</td>
<td>1895.64</td>
<td>3600.01</td>
<td>49.87%</td>
</tr>
<tr>
<td>09L1C</td>
<td>2047.30</td>
<td>3600.01</td>
<td>50.38%</td>
</tr>
<tr>
<td>09L2R</td>
<td>1716.89</td>
<td>3600.01</td>
<td>52.50%</td>
</tr>
<tr>
<td>09L2C</td>
<td>1845.72</td>
<td>3600.01</td>
<td>52.13%</td>
</tr>
</tbody>
</table>
a Gap = (Objective bound – Current Incumbent Solution) / Current Incumbent Solution (%)  
b Gap = (Objective form the solver – Best objective from GVNS) / Best objective from GVNS (%)  
c Gap = (Objective form the solver – Best objective from VNS) / Best objective from VNS (%)  

Table 2-5 Comparison of GVNS and VNS against Solver (large size instances)

<table>
<thead>
<tr>
<th>Instances</th>
<th>Solver (Gurobi)</th>
<th>GVNS</th>
<th>VNS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Obj. Value</td>
<td>Time (s)</td>
<td>Gap(^a)</td>
</tr>
<tr>
<td>10L1R</td>
<td>2811.40</td>
<td>3600.02</td>
<td>57.38%</td>
</tr>
<tr>
<td>10L1C</td>
<td>3511.14</td>
<td>3600.02</td>
<td>54.01%</td>
</tr>
<tr>
<td>10L2R</td>
<td>2004.99</td>
<td>3600.02</td>
<td>58.75%</td>
</tr>
<tr>
<td>10L2C</td>
<td>2360.74</td>
<td>3600.02</td>
<td>54.21%</td>
</tr>
<tr>
<td>11L1R</td>
<td>3459.35</td>
<td>3600.05</td>
<td>58.24%</td>
</tr>
<tr>
<td>11L1C</td>
<td>3941.44</td>
<td>3600.03</td>
<td>56.81%</td>
</tr>
<tr>
<td>11L2R</td>
<td>2818.14</td>
<td>3600.03</td>
<td>59.66%</td>
</tr>
<tr>
<td>11L2C</td>
<td>3160.38</td>
<td>3600.02</td>
<td>58.49%</td>
</tr>
<tr>
<td>12L1R</td>
<td>-</td>
<td>3600.00</td>
<td>-</td>
</tr>
<tr>
<td>12L1C</td>
<td>-</td>
<td>3600.00</td>
<td>-</td>
</tr>
<tr>
<td>12L2R</td>
<td>3748.29</td>
<td>3601.02</td>
<td>67.00%</td>
</tr>
<tr>
<td>12L2C</td>
<td>3852.16</td>
<td>3600.03</td>
<td>62.01%</td>
</tr>
<tr>
<td>13L1R</td>
<td>-</td>
<td>3600.00</td>
<td>-</td>
</tr>
<tr>
<td>13L1C</td>
<td>-</td>
<td>3600.00</td>
<td>-</td>
</tr>
<tr>
<td>13L2R</td>
<td>-</td>
<td>3600.00</td>
<td>-</td>
</tr>
<tr>
<td>13L2C</td>
<td>-</td>
<td>3600.00</td>
<td>-</td>
</tr>
<tr>
<td>14L1R</td>
<td>-</td>
<td>3600.00</td>
<td>-</td>
</tr>
<tr>
<td>14L1C</td>
<td>-</td>
<td>3600.00</td>
<td>-</td>
</tr>
<tr>
<td>14L2R</td>
<td>-</td>
<td>3600.00</td>
<td>-</td>
</tr>
<tr>
<td>14L2C</td>
<td>-</td>
<td>3600.00</td>
<td>-</td>
</tr>
<tr>
<td>15L1R</td>
<td>-</td>
<td>3600.00</td>
<td>-</td>
</tr>
<tr>
<td>15L1C</td>
<td>-</td>
<td>3600.00</td>
<td>-</td>
</tr>
<tr>
<td>15L2R</td>
<td>-</td>
<td>3600.00</td>
<td>-</td>
</tr>
<tr>
<td>15L2C</td>
<td>-</td>
<td>3600.00</td>
<td>-</td>
</tr>
</tbody>
</table>

\(^a\) Gap = (Objective bound – Current Incumbent Solution) / Current Incumbent Solution (%)  
\(^b\) Gap = (Objective form the solver – Best objective from GVNS) / Best objective from GVNS (%)  
\(^c\) Gap = (Objective form the solver – Best objective from VNS) / Best objective from VNS (%)
Table 2-6 represents the average gap and run time by group of instances. The proposed algorithms clearly performed well in finding the optimal solution. Compared with GVNS, VNS found the solution in a shorter computational time. The results obtained from large size instances demonstrated that the two algorithms found better solutions than the solver. Although GVNS was observed to have a greater computational burden than VNS, it performed better in terms of the quality of solutions. Hence, the following series of analyses are performed by GVNS.

Table 2-6 Summary of performance comparison

<table>
<thead>
<tr>
<th>Set of instances (number of instances)</th>
<th>Avg. Gap of Objective Value</th>
<th>Avg. Run Time (sec.)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Solver/GVNS</td>
<td>Solver/VNS</td>
</tr>
<tr>
<td>1-5(20)</td>
<td>0.00%</td>
<td>0.00%</td>
</tr>
<tr>
<td>6-9(16)</td>
<td>0.00%</td>
<td>0.00%</td>
</tr>
<tr>
<td>10 (4)</td>
<td>0.71%</td>
<td>-0.14%</td>
</tr>
<tr>
<td>11 (4)</td>
<td>1.35%</td>
<td>-2.20%</td>
</tr>
<tr>
<td>12 (4)</td>
<td>8.45%</td>
<td>3.59%</td>
</tr>
<tr>
<td>13 (4)</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>14 (4)</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>15 (4)</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

**Evaluation of objectives**

The two operational strategies incorporated in the container routing and scheduling problem have two conflicting objectives, travel distance and operation time. Street turning operations can help to reduce travel distance and operation time by skipping visiting the container yard, but at the same time they can yield increases in operation time due to waiting time to travel directly between two customers who have a certain time window. Similarly, while decoupling operations significantly cut down waiting time for container processing at
a customer, they might aggravate travel distance when a truck visits another customer and returns to the customer during the process. Therefore, the weights for total travel distance and operation time are very important in order to plan proper routes and schedules for a trucking company. Such weights can be determined by a decision maker that considers the monetary values and their importance. In this experiment, by setting different weights, we were able to evaluate how different objectives affect travel distance, operation time, and truck operation associated with the two strategies.

With the weights previously used, let us suppose the two extreme cases in which only travel distance and only operation time are considered. Let Obj 1 represent the objective weighted by \( \alpha_1 = 100, \alpha_2 = 1, \alpha_3 = 0 \), Obj 2 have the objective weighted by \( \alpha_1 = 100, \alpha_2 = 0, \alpha_3 = 35/60 \), and Obj 3 denote the objective weighted by \( \alpha_1 = 100, \alpha_2 = 1, \alpha_3 = 35/60 \). Figure 2-12 shows that the different objectives yield different routes and schedules even in a very small size problem. This instance is composed of three physical locations and six tasks, which means that all customers have two consecutive tasks. In order to satisfy the respective objectives, each solution applied different operational strategies for each customer. In other words, while all three cases used a street turning operation between “epfd2” and “ed1,” each objective has different task pairs executed by decoupling operations: “ed2-fped2” for Obj 1; “ed2-fped2,” “ed1-fped1,” and “fd1-epfd1” for Obj 2; and “ed2-fped2” and “fd1-epfd1” for Obj 3.
Figure 2-12 Examples of optimal routes and schedules by objective (instance 4L2R)

These different results were observed across all test instances, as shown in Figure 2-13. As expected, the travel distance of Obj 1 was less than that of Obj 2 and Obj 3; and operation time of Obj 1 was higher than that of Obj 2 and Obj 3. A similar pattern was observed in the results of Obj 3 against Obj 2, but the influence of the weights was lower than in the former two comparisons. In location set 1, instances with randomly distributed customers were more influenced by the weights compared with those with clustered customers. The opposite trend was observed in location set 2. Instances having location set 1 for randomly distributed customers and location set 2 for clustered customers have a relatively large discrepancy between objectives in terms of travel distance and operation time. Note that most of the solutions used the same number of vehicles even if different
objectives were applied, and marginal differences in vehicles used were observed for several test instances.

Figure 2-13 Comparison of % difference of total operation time and total travel time among objectives (box plot)

Figure 2-14 represents the percentage of operational strategies performed in large size test instances according to three objectives. This percentage was calculated as the number of arcs performed upon by an operation strategy divided by the number of potential arcs for this strategy. For example, if a solution has “ep1-ed1,” “ep2-fd1,” and “ep3-D,” the percentage of street turning operations performed is 33.3%. These graphs show that the weights of the objectives have a small impact on street turning operation. The reason for this result is that street turning operation is restricted by a specific pair of customers (i.e.,
empty pickup customer linked to empty delivery customers), and this pair is constrained by corresponding time windows as well. Thus, most of the arcs available for street turning without violating time constraints were utilized in all cases. On the contrary, decoupling operations were significantly affected by the weights. Overall, when the objective incorporates operation time, decoupling operation is more frequently implemented in order to reduce this factor.

![Graph showing percentage of performed strategies by objective](image)

**Figure 2-14** Comparison of percentage of performed strategies by objective

**Evaluation of operational strategies**
In the above experiments, all customers were assumed to be capable of accepting the two operational strategies under investigation. However, this assumption might not pertain in real-world problems, where customers’ requirements for sharing two resources can vary. For example, a customer may not want to share its container with others, which restricts street turning; or a customer requests that a truck stay with the container during its process to maintain security and quality of service, which prohibits decoupling operation. In order to reflect the different requirements of customers, the proposed problem is designed to allow each customer to be able to have different operational strategies through the modification of the routing network with merged nodes, as addressed in the problem description. This enables a decision maker to easily apply the proposed problem and algorithm to various scenarios, which is more realistic in practice. In order to confirm how the proposed problem performs under heterogeneous customer requirements, a series of scenarios was created based on instance sets 9 and 13 by changing customers’ participation rate to two operation strategies from 0% to 100% in 20% increments.

In this experiment, we observed that the instances with lower participation rates could be solved more quickly than ones with higher rates, because the size of the problem decreased with the number of customers that do not allow decoupling operations. Figure 2-15 represents the objective value of each instance according to the participation rate. The results show clearly that the higher participation rate can significantly reduce the operating costs. The amount of cost savings was observed to be lower in location set 1 than in location set 2. The reason is that in location set 1, a truck might have to travel a relatively long distance to pick up and deliver empty containers at the container yard—which was located
at the lower left corner of the area, close to the intermodal port—when the empty container pickup or delivery customers do not allow street turning operation.

![Graph showing objective value by customer participation rates](image)

Figure 2-15 Examples of objective value by customer participation rates in the two operational strategies

This experiment can be extended to evaluate the benefit of the two operational strategies. All 60 test instances were solved by restricting one or both of the strategies. In other words, three scenarios were generated: the problem without the two operations, with
only street turning operation, and with only decoupling operation. Based on the problem of truck operations without two strategies, the rest of problems including the original problem (considered the two strategies together) were compared. The benefit of each strategy was measured by the reduction of the objective achieved in each scenario problem against the do-nothing scenario (i.e., fleet operations restricted two strategies). Figure 2-16 presents a box plot of the objective reductions for all test instances of each scenario problem. Overall, medians of 4.2%, 20.2%, and 28.4% were obtained from the problem with only street-turn operations, with only decoupling operation, and with two operations simultaneously, respectively. The results show that the implementation of the two operations is clearly beneficial in terms of reducing travel distance and operation time. In addition, these repent decoupling operations are more helpful in reducing operating costs than street turning operations.

Figure 2-16 Box plots of objective reductions of feet operations with two strategies against fleet operations without them
The objective reductions were analyzed by a group of instance sets as shown in Figure 2-17. Given the same location of the container yard, instances having randomly distributed customers displayed a relatively high reduction of costs when the two operations are applied together. Especially, decoupling operation can lead to more cost reduction. Meanwhile, the effect of street turning operation was significantly greater in instances where the container yard is located close to the intermodal port.

![Graphs showing objective reductions](image)

**Figure 2-17** Comparison of objective reductions of feet operations with two strategies against fleet operations without them by test instance
2.6 Conclusion

This study proposed a container truck routing and scheduling problem incorporating two operational strategies, which allows a trucking company to operate a set of trucks under a shared resource environment. In contrast with a general full-truckload routing problem, a decision about two operations at a customer is made while constructing a set of routes, which is a matter of optimization. In this problem, each pickup and delivery of loaded and empty containers was considered as an independent task from the perspective of the customer’s request. By allowing street turning and decoupling operations simultaneously, this study modified a routing network to simplify the problem and formulated a mathematical model as an am-TSPTW including precedence constraints for two consecutive tasks. At the network level, site dependencies of the two strategies were considered.

This study proposed a heuristic approach that consisted of an insertion heuristic for constructing an initial solution and a modified VNS and GVNS for improving the solution. For these heuristic algorithms, a novel feasibility check algorithm was developed by considering temporally dependent time windows. The results highlighted that GVNS performed better to solve the problem within a reasonable computational time than VNS, even though VNS can find the same solution more quickly than GVNS for small size problems. Through a series of experiments, we confirmed that both operations can help to reduce the objective value. According to the location of the yard and the distribution of customers, the effect of each strategy can vary.
If resources are shared in a collaborative way among trucking companies, these two strategies are expected to provide significant benefits. These benefits might be maximizing an individual company's profit, while reducing traffic congestion, port congestion, and emissions. In addition, sharing resources can help to accomplish the public agency goals of improving freight efficiency and reducing the negative impacts of freight movement.
3.1 Introduction

The previous chapter developed the container truck routing and scheduling problem with resource sharing between customers that was facilitated by street turning and decoupling operation strategies. The main assumption in this problem was that customers allow containers and trucks assigned to them to be shared. This sharing helps to reduce the operating cost compared with truck operations without resource sharing (i.e., without the two operational strategies). The problem was designed to find an optimal set of routes to complete tasks assigned to a trucking company, thus resource sharing only occurs within the company. However, if multiple carriers in the same area adopt the underlying concept, they can exchange their own tasks with each other in a way that reduces their operating costs, which in general is called collaborative logistics. Research on collaborative logistics and associated VRP have received much attention in recent years as the transportation industry has become more competitive (Liu et al., 2010; Verdonck et al., 2013; Wang and Kopfer, 2014; Defryn, Sørensen and Cornelissens, 2016). This approach can significantly improve the efficiency and performance of logistics systems.

By adopting collaboration with regard to carrier operations, the container truck routing and scheduling problem proposed in the previous chapter can be extended. Carrier collaboration can be categorized into two types: sharing tasks and sharing vehicle capacities (Verdonck et al., 2013). While tasks requested by customers belonging to different carriers can be exchanged under the approach of sharing tasks, utilization of vehicle capacity is
increased across carriers by sharing vehicle capacities. Since the container truck movement is defined as a non-consolidated transportation problem, this study focuses on the sharing tasks method. To facilitate sharing tasks with carriers, an appropriate technology such as a web platform allowing communication between carriers should be supported. With the incredible growth of web usage, shippers can easily post their tasks on the web and a carrier can search customers and select particular assignments from the common customer pool (Archetti et al., 2009). In a container trucking industry, information exchange systems (e.g., virtual yards) have been proposed to exchange empty container tasks for street turning operations (Port of Long Beach, 2008). Through this system, a trucking company is able to post its empty container tasks and search ones that other carriers have available. If a trucking company reduces the movements related to empty containers, it can improve its efficiency and profit, because moving empty containers does not generate revenue, but creates extra trips. Therefore, the problem proposed in this chapter assumes that a trucking company exchanges empty container pickup requests with other trucking companies that participate in collaborative operations. In other words, some of the empty pickup tasks from other carriers can be selected to construct routes, and some of its own empty pickup tasks can be dropped from the list of tasks and be assigned to other carriers to minimize its operating costs. Doing so, a carrier can preserve profitable tasks such as loaded container pickup and delivery tasks and reduce travel miles produced by empty container movements. Since sharing resources between carriers focuses only on empty pickup tasks, which are non-revenue, complex negotiations, auction mechanisms, and profit sharing would not need to be considered in this problem. By exchanging customers’ requests, the overall travel miles
of empty container trucks in local areas can decrease, which results in reduced traffic congestion, port congestion, and emissions as well.

A decision on selecting empty pickup tasks will be made during construction of routes by considering constraints. In addition, to simplify the problem and its size, all customers whose tasks are identified for street turning and decoupling operations allow these two operations, and a limited number of empty container pickup tasks from other carriers are considered in the problem. Typically, the more carriers participate, the better solutions the problem will find. However, the number of customers has a significant impact on increasing the computation time. Hence, we assume that the problem ensures a sufficient number of participating carriers; thus, empty pickup tasks that belong to a carrier, but are not selected by the carrier, can be assigned to outside carriers, while only some of an abundance of tasks from outsiders in the system are selected in advance and included in the optimization problem.

3.2 Literature review

The problem proposed in this chapter is designed to exchange empty container pickup requests between carriers participating in collaborative operations. Although the system has multiple carriers, the routing and scheduling is planned from the perspective of an individual trucking company. Hence, this problem adopts the idea that a trucking company can perform a subset of selected tasks and outsource the remaining tasks. At the same time, it can accept some tasks outsourced from outside carriers. Assuming that only empty container pickup tasks can be shared with each other, a trucking company includes
selected empty container pickup tasks in its routes by solving the routing and scheduling problem, while it should perform all other tasks. Therefore, this problem can be formulated as a variant of the selective vehicle routing problem (SVRP).

The problems determining the sequence of customers served by vehicle(s) and customers to be served in route(s) simultaneously are defined as a class of VRPs with profit (Archetti, Speranza and Vigo, 2014). Compared with the extensive research on VRP, there is little literature addressing VRPs with profit, a survey of which was provided by Archetti et al. (2014). When the problem considers the route of a single vehicle, it is defined as the traveling salesman problem (TSP) with profits that aims to maximize profits or minimize costs through selectively visiting customers. By varying the objective and constraints, the TSP with profits has been modified. Feillet et al. (2005) classified this problem into three types: 1) the profitable tour problem (PTP) where the objective is to minimize travel costs minus collected profits from the visited customers, 2) the orienteering problem (OP), which is also called the selective traveling salesman problem (STSP) (Laporte and Martello, 1990), which aims to maximize profit constrained by upper bound travel costs, and 3) the prize-collecting TSP (PCTSP), which finds an optimal route that minimizes costs subject to lower-bound profits. Feillet et al. (2005) also provided integer linear programming formulations according to the classification.

The TSP with profits has been extended to the VRP with profits, which is called the team orienteering problem (TOP), by considering multiple vehicles in the problem (Chao, Golden and Wasil, 1996; Archetti et al., 2009; Bouly, Dang and Moukrim, 2010). Archetti et
al. (2009) proposed two problems related to the TOP. One is the capacitated team orienteering problem (CTOP), which aims to maximize the total collected profit subject to the capacity and maximum task duration of the vehicles; another is the capacitated profitable tour problem (CPTP), the objective of which is maximizing the total marginal profit (i.e., the total collected profit minus travel cost) with vehicle capacity constraints. Two tabu search algorithms and a VNS algorithm were developed to solve the proposed problems. Bouly et al. (2010) also studied a TOP to maximize the profit with a given number of vehicles subject to the maximum task duration of vehicles and proposed a memetic algorithm. By incorporating multiple-depot VRP, Aras et al. (2011) extended the TOP to model a reverse logistics problem of a firm that collects used products from its dealers. They called this the selective multi-depot vehicle routing problem with pricing. They formulated two mixed-integer linear programming models based on TOP that aim to maximize the difference between revenue and cost associated with visiting dealers to collect products and developed a tabu search heuristic as a solution approach.

Several studies have embedded time window constraints widely considered in the VRPTW into the OP (e.g., Righini and Salani, 2009) and TOP (e.g., Vansteenwegen et al., 2009). To solve this type of problem efficiently, various solution approaches have been proposed, for instance, a dynamic programming algorithm (Righini and Salani, 2009), an iterated local search algorithm (Vansteenwegen et al., 2009), a simulated annealing heuristic (Lin and Yu, 2012), an artificial bee colony algorithm (Cura, 2014), a VNS (Tricoire et al., 2010), and a linear programming-based granular variable neighborhood search (Labadie et al., 2012).
The main assumption in the VRP with profits is that the profit of each customer is known a priori, and customers to be served by vehicles are then selected based on these values. However, the problem proposed in this chapter only considers selecting empty container pickup tasks, which are non-profit exercises. The profitable tasks should be completely served by the company without sharing resources between carriers. Hence, the problems for the VRP with profits could not be directly utilized. However, the underlying idea in the SVRP—that is, selecting customers utilizing cost structures—can be applied to our case, which has been adopted for several collaborative VRPs. Instead of profits collected from the customers visited, Defryn et al. (2016) considered the penalized costs to the unvisited customers; thus, the objective is to minimize the travel costs and the penalized costs such that each tour length is not over its target maximum. By solving the problem with the randomized, multi-start variable neighborhood search algorithm, a set of routes was found with selection of customers to be served. After that, the method applied a cost allocation routine to split the costs incurred among the participating carriers.

A different approach addressing the collaborative VRP was proposed by Bolduc et al. (2006) and Bolduc et al. (2007). They proposed a vehicle routing problem with private fleet and common carrier (VRPPC), in which some customers were selected to be served by a common carrier with predefined costs. This concept is similar to the outsourcing option in VRP. They proposed a heuristic algorithm comprising selection, routing, and improvement steps. Côté and Potvin (2009) elaborated the same problem and developed a tabu search heuristic. By extending this problem, Liu et al. (2010b) considered bidirectional task
selection relationships between a private fleet and an external carrier. In addition to allocating some customers to the external carrier, the private fleet also can serve some tasks belonging to the external carrier, and thus the objective included compensative payments gained from external tasks and penalty costs incurred from private tasks. Since they formulated the problem for full truck loads along with a cost structure for exchanging tasks between carriers, their work is closely related to the proposed problem in this chapter.

The proposed container truck routing and scheduling problem with collaboration thus adopts the problem proposed by Liu et al. (2010b). However, our approach is distinct from their work. In this problem, multiple carriers are assumed to be able to participate in collaboration and empty container pickup tasks can be allocated to one of several carriers. Without loss of generality, the task exchanging cost is assumed to be equal across all carriers, because moving empty containers is non-revenue, thus the same costs term used in Liu et al. (2010b) is applicable. In addition, the proposed problem only considers selective empty containers. Assuming a sufficient number of tasks posted on the web platform, these can be easily transferred to an appropriate outside carrier. Problems considering selective pickup have been studied in literature on pickup and delivery (Gribkovskaia, Laporte and Shyshou, 2008; Ting and Liao, 2013). Ting and Liao (2013) pointed out that this problem can substantially shrink the transportation cost by focusing on fulfilling delivery demands. Finally, the proposed problem considered distinct features of container movements addressed in the previous chapter and associated operational strategies with time constraints.
3.3 Modeling the problem

This section describes the problem of optimizing truck routes and schedules for moving a set of containers in a trucking company and exchanging a certain type of tasks with other companies. For this problem, it is assumed not only that many trucking companies are willing to participate in horizontal collaborations, but also that all participants operate their fleets by fully employing the street turning and decoupling operations addressed in the previous chapter. This helps to improve the efficiency of overall container truck movements in the collaboration, as well as that of routing plans for an individual company. The problem is set on a general graph $G(N, A)$, where $N$ is the set of nodes and $A$ is the set of arcs, which reflect the distinctive characteristics of container movements as mentioned in Chapter 2. A systemic platform for exchanging tasks required for this problem is explained. The proposed mathematical formulation minimizes the operation costs, incorporating task exchange costs that are incurred by selection of tasks, particularly empty container pickup tasks.

3.3.1 Problem description

In a collaborative environment, a trucking company makes a daily routing plan for moving two types of containers, loaded containers and empty containers, with tasks shared with other companies. From the point of view of an individual company, it wants to reduce its operating costs, in particular, those caused by empty container pickup requests, because such requests do not generate revenue. The proposed problem aims to find an optimal route plan for a trucking company that is able to select and exchange empty container pickup
requests with other carriers. Here, a carrier that wants to have a operation plan is called the *private carrier*, while other carriers in the system are called *outside carriers*.

In similar fashion to the base model for container truck routing and scheduling problems, the problem is composed of one depot, one intermodal terminal, one container yard, and six types of customers (i.e., $C_{fd}, C_{fp}, C_{ed}, C_{ep}, C_{fped}, C_{epfd} \in C$). In addition to empty container pickup customers of the private carrier, those of the outside carriers, $C_{ep}^{ps}$, are included in this problem. To simplify the notation, $C_{ep}^{ps}$ is added into $C_{ep}$. Each customer $i$ has a time window $[a_i, b_i]$, service time $s_i$, container process time (i.e., packing and unpacking time) $p_i$, and carrier information $z_i$, where $z_i = 1$ if customer $i$ is in the private carrier, otherwise $z_i = 0$. Compared with the predetermined time window for customers in $\{C_{fd}, C_{fp}, C_{ed}, C_{ep}\}$, the problem considers flexible time windows for customers in $\{C_{fped}, C_{epfd}\}$ to allow decoupling operations. The underlying structure of the container truck routing network in this problem is similar to the network described in Chapter 2. The difference is that empty pickup customers $C_{ep}$ and $C_{epfd}$ can be selectively visited, while the other customers are required to be served by the private carrier. When the selection of empty pickups happens, this will incur the cost involved, which is defined as the task exchanging cost $e_i$ in this study. In other words, if an empty pickup task of the private carrier is not selected, it should be allocated to one of the outside carriers in collaboration, and the private carrier should pay the task exchanging cost to the carrier; whereas, when the private carrier serves an empty pickup task from outside carriers, it should receive the cost from them. Hence, the problem includes these costs in the objective function.
A task-exchanging system is assumed to have a sufficient number of participants and associated tasks and should ensure that unselected empty pickup tasks of the private carrier are assigned to outside carriers. Figure 3-1 represents the flow chart of the proposed problem supported by the task-exchanging system. A web-based exchange system has been proposed and implemented in collaborative logistics (Naja, Eshghi and Dullaert, 2013; Zolfagharinia and Haughton, 2014). Through such a system, the private carrier can easily find the candidate empty pickup tasks and other carriers that are willing to receive its tasks. Even though the inclusion of all tasks posted on the system helps to find the best empty pickup tasks from outside carriers, the computational burden of creating a routing model is significantly sensitive to the problem size. To avoid the routing problem becoming computationally expensive, a pre-selection algorithm is introduced. Through this step, several tasks that are expected to improve the efficiency of routes are selected in advance. Given a set of tasks including pre-selected ones, the routing and scheduling problem is solved to find an optimal selection of tasks and routes simultaneously.
Figure 3-1 Overall process of the container truck routing problem with collaboration

Figure 3-2 represents how the proposed routing problem with collaboration works. In a general problem (Figure 3-2a), a set of routes is constructed by solving the problem proposed in Chapter 2. In the proposed collaboration problem, the web-based exchange system with the pre-selection step can provide the set of empty pickup tasks (Figure 3-2b). By combining these into the given tasks in the private carrier, the problem can find better routes with a consideration of the selection of shared empty pickups.
3.3.2 Mathematical formulation

The proposed mathematical formulation uses suitable modifications of the am-TSPTW presented in Chapter 2 to meet the requirements of the selective VRP. The key modification is relaxing the constraints of visiting all customers by allowing selectivity of empty container pickups. Additionally, the cost structure of exchanging tasks is incorporated within the optimization problem. From the perspective of creating a routing plan for an individual carrier, Liu et al. (2010b) formulated the minimization of the carrier’s total cost including penalty costs to an outside carrier and compensative costs acquired from the outside carrier. In a similar fashion, the objective function considered in the proposed problem is to minimize the operating costs that are composed of the number of trucks used, total travel distance traveled, total truck operation time, and task exchanging costs, with respective weights.
\[
\min Z = \alpha_1 \sum_v \sum_j x_{ij}^v + \alpha_2 \sum_v \sum_{ij} c_{ij} x_{ij}^v + \alpha_3 \sum_v T_{n+1}^v - T_0^v + \alpha_4 \sum_i e_i (z_i - y_i)
\] (3.1)

Subject to
\[
\sum_v \sum_j x_{ij}^v = y_i, \ \forall \ i \in C
\] (3.2)
\[
y_i = 1, \ \forall \ i \in \{C_{fd}, C_{fp}, C_{ed}, C_{fped}\}
\] (3.3)
\[
y_i \leq 1, \ \forall \ i \in \{C_{ep}, C_{epfd}\}
\] (3.4)
\[
\sum_j x_{ij}^v - \sum_j x_{ji}^v = 0, \ \forall \ i \in C, \ \forall \ v \in V
\] (3.5)
\[
\sum_j x_{0j}^v \leq 1, \ \forall \ v \in V
\] (3.6)
\[
\sum_j x_{nj}^v - \sum_j x_{0j}^v = 0, \ \forall \ v \in V
\] (3.7)
\[
T_i^v + S_i + t_{ij} - T_j^v \leq M(1 - x_{ij}^v), \ \forall \ i,j \in N, \ \forall \ v \in V
\] (3.8)
\[
T_i^v + S_i + P_i(1 - x_{ij}^v) + t_{ij} x_{ij}^v \leq T_j^v + M(1 - y_i), \ \forall \ (i,j) \in A_c, \ \forall \ v \in V
\] (3.9)
\[
a_i \sum_j x_{ij}^v \leq T_i^v, \ \forall \ i \in N, \ \forall \ v \in V
\] (3.10)
\[
T_i^v \leq b_i \sum_j x_{ij}^v, \ \forall \ i \in N, \ \forall \ v \in V
\] (3.11)
\[
T_{n+1}^v - T_0^v \leq W, \ \forall \ v \in V
\] (3.12)

where
\[
x_{ij}^v: 1 \text{ if vehicle } v \text{ travels from node } i \text{ and node } j, \text{ otherwise } 0
\]
\[
y_i: 1 \text{ if task } i \text{ is served, otherwise } 0
\]
\[
c_{ij}: \text{Travel distance from node } i \text{ and node } j
\]
\[
t_{ij}: \text{Travel time from node } i \text{ and node } j
\]
\[
e_i: \text{Task exchange cost at node } i
\]
$z_i$: 1 if task $i$ belongs to a company, otherwise 0

$V$: Set of vehicles

$D$: Set of depot, \{0, $n$ + 1\}

$C$: Set of customers \{1, 2, ..., $n$\}

$N$: Set of nodes, $D \cup C$

$A_c$: Set of pairs of two consecutive tasks

$T_i^v$: Service start time at node $i$ by vehicle $v$

$S_i$: Duration for service at node $i$

$P_i$: Duration for packing or unpacking of container at node $i$

$a_i, b_i$: Lower and upper time window at node $i$

$W$: Maximum work-shift hours for driver

$\alpha_1, \alpha_2, \alpha_3, \alpha_4$: Weighted factors

$M$: Arbitrary Large number

Constraints (3.2) ensure that every selected node be visited exactly once in all routes. Constraints (3.3) and (3.4) ensure that customers who are not empty pickups should be served, but empty pickup customers can be selectively visited. Constraints (3.5) represent flow conservation for each tour. Constraints (3.6) allow that vehicles can be used for the route at most once. Constraints (3.7) dictate that each vehicle used should return to the depot after visiting customers. Constraints (3.8) calculate the service start time at each customer. Constraints (3.9) enforce the beginning of the time window for the consecutive tasks $C_{fped}$ and $C_{epf_d}$, computed by adding the container process time and pickup and drop off time to the service start time of the preceding task. In these constraints, service time is
conditionally considered. Constraints (3.10) and (3.11) are time windows. Constraints (3.12) limit the maximum drivers’ work-shift hours for each route.

3.4. Solution approach

This section describes a pre-selection algorithm that determines a small set of candidate empty container pickup tasks obtained from outside carriers, followed by modification of the GVNS developed in the previous chapter to solve the defined mathematical formulation. As previously mentioned, since am-TSPTW is NP-hard, the container truck routing problem with selective empty container pickups is also an NP-hard problem. Literature on this class of SVRPs as well as VRPs has proposed numerous heuristic and meta-heuristic approaches in order to solve the problem within reasonable computation times. Our results in the previous chapter also showed that the two-stage solution approach based on insertion and GVNS algorithm was able to solve the proposed container truck routing and scheduling problem. However, as the problem size increases, the computational burden increases rapidly, due to the increasing complexity of the problem with temporal dependencies between tasks. Adding more outside carriers’ tasks into the problem aggravates the computational burden even further with the meta-heuristic approach. In order to reduce the size for the routing and scheduling problem, the pre-selection algorithm is applied. After this step, the two-stage algorithm solves the problem. The algorithm is modified by allowing it to handle the selection of tasks.

3.4.1. Pre-selection algorithm
A simple algorithm is proposed in order to prevent all outside carriers’ tasks being included in the routing model. In terms of empty container movements, directly connecting an empty container pickup node with an empty delivery node is a way to reduce the additional trips required to visit the container yard, as addressed in street turning operations. Assuming that this is allowed, we expect that a route tries to select an empty pickup node close to an empty delivery node if the problem has multiple empty pickup nodes. Doing so, the route can reduce total travel distance and operation time. Hence, we suppose that an empty pickup node close to empty delivery nodes might have a higher chance to be exchanged with the existing empty pickup nodes from the private carrier. Also, the empty pickup node will be linked from a location that might be the depot, container yard, or loaded container delivery customers’ destination in the modified network. Thus, the various travel distances from a customer to empty containers passing through empty pickup locations from outside carriers need to be considered in order to select among them. Using this underlying idea, the algorithm can pre-select relatively important empty pickup nodes among all tasks.

The importance can be measured by a centrality indicator. Several indicators have been proposed to address the importance of nodes in graph theory and network analysis (Boldi and Vigna, 2014). By examining the value of the indicator for all nodes in the network, we can identify which node is the most influential node to others. By adopting this concept, the algorithm proposed in this study applies harmonic centrality, which is one of the indicators based on the harmonic mean of all distances from a node. The reason for using this centrality is that in our problem, only one pair of nodes is finally selected in the routing problem, so that if an empty pickup node is very close to a certain empty delivery node and
its possible preceding nodes, that node should be more heavily weighted. In general, the harmonic mean is useful when particularly small values are critical. Closeness centrality for empty container pickup nodes can be measured as follows:

\[ CC_j = \sum_{i \in \{Y, T, D, C_{fd}\}} \frac{1}{d(i, j, k)}, \quad i \in \{Y, T, D, C_{fd}\}, \quad j \in C_{ep}^{OC}, \quad k \in C_{ed}, \]  \hspace{1cm} (3.13)

where \( CC_j \) is closeness centrality for node \( j \); \( d(i, j, k) \) denotes the travel distance from \( i \) to \( k \) through \( j \); \( C_{ep}^{OC} \) represents a set of empty container pickup customers from outside carriers; \( C_{ed} \) is a set of empty container delivery customers from the private carrier; and \( Y, T, \) and \( D \) denote the yard, terminal, and depot, respectively. Note that the higher its closeness centrality, the higher the chance of selection a node has.

The algorithm starts by calculating closeness centrality for each empty pickup customer. The customers are then sorted in descending order of closeness centrality. The algorithm returns a list of pre-selected empty pickup customers \( C_{ep}^{ps} \) with the \( n \) highest values. This approach forces additional nodes included in the routing model not to exceed a certain level. The number of pre-selected customers \( n \) is determined by the decision maker in such a way as not to worsen the computational burden of the algorithm for the routing model.

### 3.4.2. Modification to two-stage algorithm
An algorithm based on a VNS has been proposed and shown to perform well in solving the SVRP (Tricoire et al., 2010; Labadie et al., 2012). Thus, the two-stage algorithm composed of insertion heuristic and GVNS proposed for the base model is used to solve the problem with appropriate modifications.

For initial solution construction, the insertion heuristic is used with all customers in the base model. However, the heuristic creates an initial solution without taking into account the outside carriers’ tasks (i.e., \(C_{ep}^{ps}\)) by keeping the simplicity of the heuristic. In other words, the algorithm starts with a given set of customers, \(\{C_{fd}, C_{fp}, C_{ed}, C_{ep}, C_{fped}, C_{epfd}\}\), where \(C_{ep}\) does not include \(C_{ep}^{ps}\), and all the customers should be inserted in the routes. The selection of tasks is not considered at this stage, but is handled in the following improvement step. Since the problem will search a better solution than one containing all tasks of the private carrier by selecting and exchanging tasks, it makes sense to find an initial solution only with tasks belonging to the private carrier.

Previously, the GVNS algorithm was developed to improve an initial solution with eight neighborhood structures: Intra-route insertion \(N_1\), Inter-route insertion \(N_2\), Intra-route double insertion \(N_3\), Inter-route double insertion \(N_4\), Intra-route swap \(N_5\), Inter-route swap \(N_6\), Intra-route 2-opt \(N_7\), and Inter-route 2-opt \(N_8\). These operators cannot generate a neighborhood solution involving selection of nodes. Therefore, the following neighborhood structures can be added to the existing ones, allowing movement for task selection.
- Selectable customer add $N_9$ (Figure 3-3a): A customer is randomly selected from the selectable customer set and inserted to a random location of a random route.

- Selectable customer drop $N_{10}$ (Figure 3-3b): A customer who is eligible for the selection is identified from a certain route and extracted from its current location. This customer is added into the selectable customer set for later.

- Selectable customer swap $N_{11}$ (Figure 3-3c): Two customers who can selectively be in the routes are swapped, where one is in a route and the other is in the selectable customer set.

![Diagram of neighborhood structures]

Figure 3-3 Additional neighborhood structures
With these additional neighborhood structures, randomization of the set of neighborhood structures is introduced into GVNS. While a determined set of neighborhood structures was used in the previous GVNS, here a new set of neighborhood structures is generated by randomly changing the order at every iteration and applying both GVNS and its local search method, VND. As shown in Figure 3-4, the two processes are added into the algorithm as follows:

\[ N_k \leftarrow f_1(N_k) = \text{random sample}(\{N_2, N_4, N_6, N_8\}, 2) + \text{random sample}(\{N_9, N_{10}, N_{11}\}, 3) \]

\[ N_i \leftarrow f_2(N_i) = \text{random sample}(\{N_2, N_4, N_6, N_8\}, 1) + \text{random sample}(\text{remainings}) \]

where \text{random sample}(A, n) is a function randomly generating \( n \) elements of \( A \). Since the VNS and VND heuristic is designed to revisit the first structure when the solution is improved, it is recommended that the first structure is one having small movement. Therefore, in the randomized function for the structure set, the first choice is restricted. Also, to reduce the computation time, only five structures are used at master level with random selection of two elements from inter-route operators.
Figure 3-4 GVNS with random neighborhood structures
\subsection*{3.5 Numerical experiments}

\subsubsection*{3.5.1 Test instances}

The same 60 test instances randomly generated in Chapter 2 were utilized for the private carrier. A set of empty container pickup customers for outside carriers, $C_{ep}^{OC}$, was randomly generated in the range [10,10] and [40,40] on the same Euclidean plane of size 50 miles by 50 miles. Initially, a total of 50 pickups for outside carriers is set. Among them, several tasks with a given size $C_{ep}^{ps}$ will be selected and added into the problem later. The beginning of the time window and the time window interval for $C_{ep}^{OC}$ are set as follows.

- The beginning of time window for $C_{ep}^{OC}$: a uniform random variable from 8 am to 12 pm corresponding to the time window for $C_{ed}$.
- The time window interval for $C_{ep}^{OC}$: a uniform random variable ranging from 60 minutes to 240 minutes in increments of 30 minutes.

Considering the number of empty container delivery customers in the private carrier, the number of empty container pickup customers in outside carriers is determined as shown in Table 3-1

\begin{table}[h]
\centering
\begin{tabular}{|c|c|c|c|c|c|c|c|c|}
\hline
Customer set & $C_{fd}$ & $C_{fp}$ & $C_{ed}$ & $C_{ep}$ & $C_{epfd}$ & $C_{fped}$ & $C_{ep}^{ps}$ & total \\
\hline
1 & 0 & 1 & 1 & 1 & 0 & 1 & 3 & 7 \\
2 & 1 & 1 & 1 & 0 & 1 & 1 & 3 & 8 \\
3 & 1 & 1 & 1 & 1 & 1 & 1 & 3 & 9 \\
4 & 2 & 0 & 2 & 0 & 2 & 2 & 3 & 11 \\
5 & 2 & 2 & 2 & 0 & 2 & 2 & 3 & 13 \\
\hline
\end{tabular}
\caption{Number of customers by customer set including empty pickups from outside}
\end{table}
<table>
<thead>
<tr>
<th></th>
<th>6</th>
<th>3</th>
<th>1</th>
<th>2</th>
<th>1</th>
<th>3</th>
<th>2</th>
<th>3</th>
<th>15</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>17</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>19</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>5</td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>5</td>
<td>4</td>
<td>5</td>
<td>25</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>5</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>7</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>7</td>
<td>4</td>
<td>10</td>
<td>40</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>5</td>
<td>7</td>
<td>10</td>
<td>3</td>
<td>5</td>
<td>10</td>
<td>10</td>
<td>50</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>12</td>
<td>8</td>
<td>8</td>
<td>2</td>
<td>12</td>
<td>8</td>
<td>10</td>
<td>60</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>15</td>
<td>10</td>
<td>15</td>
<td>5</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>90</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>20</td>
<td>20</td>
<td>15</td>
<td>10</td>
<td>20</td>
<td>15</td>
<td>15</td>
<td>115</td>
<td></td>
</tr>
</tbody>
</table>

3.5.2 Computational results

Evaluation of performance

The solver and the proposed algorithm were run on the same environment in which the experiments in Chapter 2 were conducted. As mentioned in the previous section, a randomized set of neighborhood structures was used at each iteration. The maximum number of iterations of 30 and the maximum number of non-improvements of 10 were set as the stopping criteria. The weight factors of the objective were assumed to be $\alpha_1 = 100, \alpha_2 = 1, \alpha_3 = 35/60, \text{and } \alpha_4 = 1$. We assume that the exchanging cost for each task $i, e_i$, has the same value of 10 for all tasks. In this experiment, selected empty container pickup tasks for outside carriers, $C_{ep}^{ps}$, are randomly chosen from $C_{ep}^{OC}$ without applying the preselection algorithm.

As shown in Table 3-2 and Table 3-3, the proposed GVNS performed well compared with the solver. In large size instances, the algorithm found the better solution, while the solver could not find the solution within the given time limit.
Table 3-2 Comparison of GVNS against Solver (small size instances)

<table>
<thead>
<tr>
<th>Instances</th>
<th>Solvers (Gurobi)</th>
<th>GVNS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Obj. Value</td>
<td>Time (s)</td>
</tr>
<tr>
<td>01L1R</td>
<td>446.66</td>
<td>0.05</td>
</tr>
<tr>
<td>01L1C</td>
<td>461.51</td>
<td>0.05</td>
</tr>
<tr>
<td>01L2R</td>
<td>443.63</td>
<td>0.07</td>
</tr>
<tr>
<td>01L2C</td>
<td>458.64</td>
<td>0.08</td>
</tr>
<tr>
<td>02L1R</td>
<td>700.76</td>
<td>0.61</td>
</tr>
<tr>
<td>02L1C</td>
<td>755.48</td>
<td>0.34</td>
</tr>
<tr>
<td>02L2R</td>
<td>700.76</td>
<td>0.88</td>
</tr>
<tr>
<td>02L2C</td>
<td>755.04</td>
<td>0.74</td>
</tr>
<tr>
<td>03L1R</td>
<td>579.47</td>
<td>0.65</td>
</tr>
<tr>
<td>03L1C</td>
<td>656.05</td>
<td>1.39</td>
</tr>
<tr>
<td>03L2R</td>
<td>579.47</td>
<td>4.78</td>
</tr>
<tr>
<td>03L2C</td>
<td>656.05</td>
<td>1.58</td>
</tr>
<tr>
<td>04L1R</td>
<td>763.82</td>
<td>17.53</td>
</tr>
<tr>
<td>04L1C</td>
<td>839.29</td>
<td>8.50</td>
</tr>
<tr>
<td>04L2R</td>
<td>763.82</td>
<td>69.11</td>
</tr>
<tr>
<td>04L2C</td>
<td>826.36</td>
<td>81.42</td>
</tr>
<tr>
<td>05L1R</td>
<td>1,148.37</td>
<td>1,068.50</td>
</tr>
<tr>
<td>05L1C</td>
<td>1,342.91</td>
<td>1,152.80</td>
</tr>
<tr>
<td>05L2R</td>
<td>1,110.24</td>
<td>635.62</td>
</tr>
<tr>
<td>05L2C</td>
<td>1,270.83</td>
<td>3,600.01</td>
</tr>
<tr>
<td>06L1R</td>
<td>1,116.88</td>
<td>3,600.01</td>
</tr>
<tr>
<td>06L1C</td>
<td>1,235.57</td>
<td>3,600.01</td>
</tr>
<tr>
<td>06L2R</td>
<td>1,078.04</td>
<td>3,600.02</td>
</tr>
<tr>
<td>06L2C</td>
<td>1,194.69</td>
<td>3,600.05</td>
</tr>
<tr>
<td>07L1R</td>
<td>1,156.78</td>
<td>3,600.02</td>
</tr>
<tr>
<td>07L1C</td>
<td>1,450.50</td>
<td>3,600.01</td>
</tr>
<tr>
<td>07L2R</td>
<td>1,184.47</td>
<td>3,600.02</td>
</tr>
<tr>
<td>07L2C</td>
<td>1,395.09</td>
<td>3,600.01</td>
</tr>
<tr>
<td>08L1R</td>
<td>1,433.66</td>
<td>3,600.01</td>
</tr>
<tr>
<td>08L1C</td>
<td>1,610.03</td>
<td>3,600.01</td>
</tr>
<tr>
<td>08L2R</td>
<td>1,467.68</td>
<td>3,600.01</td>
</tr>
<tr>
<td>08L2C</td>
<td>1,578.82</td>
<td>3,600.01</td>
</tr>
<tr>
<td>09L1R</td>
<td>1,669.79</td>
<td>3,600.02</td>
</tr>
<tr>
<td>09L1C</td>
<td>1,799.08</td>
<td>3,600.02</td>
</tr>
<tr>
<td>09L2R</td>
<td>1,668.63</td>
<td>3,600.02</td>
</tr>
<tr>
<td>09L2C</td>
<td>1,796.63</td>
<td>3,600.04</td>
</tr>
</tbody>
</table>

<sup>a</sup> Gap = (Objective bound − Current Incumbent Solution) / Current Incumbent Solution (%)  
<sup>b</sup> Gap = (Objective form the solver − Best objective from GVNS) / Best objective from GVNS (%)
<table>
<thead>
<tr>
<th>Instances</th>
<th>Solver (Gurobi)</th>
<th>GVNS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Obj. Value</td>
<td>Time (s)</td>
</tr>
<tr>
<td>10L1R</td>
<td>2,057.94</td>
<td>3,600.06</td>
</tr>
<tr>
<td>10L1C</td>
<td>2,538.40</td>
<td>3,600.03</td>
</tr>
<tr>
<td>10L2R</td>
<td>1,932.68</td>
<td>3,600.03</td>
</tr>
<tr>
<td>10L2C</td>
<td>2,248.12</td>
<td>3,600.03</td>
</tr>
<tr>
<td>11L1R</td>
<td>2,777.38</td>
<td>3,600.02</td>
</tr>
<tr>
<td>11L1C</td>
<td>3,111.57</td>
<td>3,600.02</td>
</tr>
<tr>
<td>11L2R</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>11L2C</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>12L1R</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>12L1C</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>12L2R</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>12L2C</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>13L1R</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>13L1C</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>13L2R</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>13L2C</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>14L1R</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>14L1C</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>14L2R</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>14L2C</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>15L1R</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>15L1C</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>15L2R</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>15L2C</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

<sup>a</sup> Gap = (Objective bound – Current Incumbent Solution) / Current Incumbent Solution (%)

<sup>b</sup> Gap = (Objective form the solver – Best objective from GVNS) / Best objective from GVNS (%)

Table 3-4. Summary of performance

<table>
<thead>
<tr>
<th>Set of instances (number of instances)</th>
<th>Avg. Gap of Objective Value</th>
<th>Avg. Run Time (sec.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>GVNS v.s. Solver</td>
<td>Solver</td>
<td>GVNS</td>
</tr>
</tbody>
</table>

93
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th>332.23</th>
<th>3.16</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-5(20)</td>
<td>0.00%</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6-9(16)</td>
<td>0.01%</td>
<td>3,600.02</td>
<td>44.22</td>
<td></td>
</tr>
<tr>
<td>10 (4)</td>
<td>0.40%</td>
<td>3,600.02</td>
<td>209.90</td>
<td></td>
</tr>
<tr>
<td>11 (4)</td>
<td>7.29%</td>
<td>3,600.02</td>
<td>417.39</td>
<td></td>
</tr>
<tr>
<td>12 (4)</td>
<td>-</td>
<td>3,600.02</td>
<td>1,012.46</td>
<td></td>
</tr>
<tr>
<td>13 (4)</td>
<td>-</td>
<td>3,600.02</td>
<td>1,283.97</td>
<td></td>
</tr>
<tr>
<td>14 (4)</td>
<td>-</td>
<td>3,600.02</td>
<td>2,977.02</td>
<td></td>
</tr>
<tr>
<td>15 (4)</td>
<td>-</td>
<td>3,600.02</td>
<td>3,028.53</td>
<td></td>
</tr>
</tbody>
</table>

**Evaluation of the problem with collaboration and pre-selection algorithm**

In order to evaluate the effect of collaboration, the proposed problem was compared with the problem without collaboration. The results for the problem without collaboration were obtained from the experiment in Chapter 2. In this experiment, the pre-selection algorithm was applied before solving the problem. The number of empty container pickup tasks from outside carrier as defined in Table 3-1 was also used to select them for each test instance. These results were also compared with those from the problem without pre-selection algorithm that were obtained from the performance evaluation.

Figure 3-5 represents objective reduction of the proposed problem with pre-selection algorithm, against the problem without collaboration and the problem without pre-selection algorithm. Results show that the truck operation with sharing of empty containers with other carriers was more efficient. Especially, the effect of collaboration was more significant in location set 1 than in location set 2. The reason for this is that in the location set 1 the container yard was located at the lower left corner of the area close to the intermodal port, so that trucks might have to travel longer distances to visit the container yard if a route cannot find customer pairs for street turning operation. Outsourcing several empty
containers and accepting them from outside carriers can help to reduce travel distances caused by empty container movements passing through the container yard.

Comparison between solutions with and without applying the pre-selection algorithm shows that the pre-selection algorithm can help to decrease the objective. However, the effect of the pre-selection algorithm was observed not to be critical, especially in test instances generated on location set 2. In the proposed algorithm, although the candidates were selected based on their distance, their time windows might have a significant influence on including them in the solution in the routing and scheduling problem. Therefore, for the improvement of this algorithm, time windows could be considered to measure the closeness centrality.

Figure 3-5 Comparison of objective reduction of problem with pre-selection algorithm
Evaluation of size of pre-selected tasks from outside carriers

The influence of the size of pre-selected tasks from outside carriers on the objective was explored. This size is highly related to the number of empty container delivery customers, since empty pickups from outside carriers are expected to be selected to find empty pickup and delivery pairs involving street-turn operation. The sizes of pre-selected tasks were determined as the percentage of the number of empty container delivery tasks in the private carrier. This experiment was applied to test instance sets 10 and 12 in which a total of five and then empty container delivery tasks exist, respectively. Figure 3-6 shows that when the size increased, the solutions were improved. Depending on the instances and associated locations, the effect of size could be marginal.

![Graph showing the effect of size on solutions]

a) Instance set 10
b) Instance set 12

Figure 3-6 Comparison of objective by size of pre-selected tasks from outside carriers

**Evaluation of exchanging cost**

The objective function in the problem includes the term of costs for exchanging tasks between the private carrier and outside carriers. Under the assumption of the same cost for all tasks, if the number of outsourced tasks is equal to the number of received tasks, no cost is incurred. If there are more outsourced tasks, the cost increases. In contrast, if there are more received tasks, the cost decreases. This cost can affect the solution along with other terms of cost.

Using instance sets 10 and 12, the effect of the exchanging cost was analyzed as shown in Figure 3-7. The cost was changed from 10 to 100. Results show that the change of the objective varies according to the exchanging cost and instance set. The reason is that with the cost structure, the selection of tasks for the solution is different. As shown in Table 3-5,
for example, in test instance 10L2C, two empty container pickup tasks belonging to the private carrier are outsourced and five empty container pickup tasks are received from outside carriers with exchanging costs of 50. On the other hand, with costs of 70 and 100, no task was outsourced, but the same amount of tasks was accepted. This resulted in a decrease in the total cost, but increases in the total travel distance and operation time.

![Graph](image)

**Figure 3-7** Comparison of objective according to the exchanging cost
Table 3-5 Comparison of number of empty pickups for collaboration by exchanging cost

<table>
<thead>
<tr>
<th>Instances</th>
<th># empty pickups in Private carrier, ${C_{ep}, C_{epfd}}$</th>
<th># empty pickup in outside carriers, $C_{ep}^{ps}$</th>
<th># empty pickups sent to outside carriers</th>
<th># empty pickups received from outside carriers</th>
</tr>
</thead>
<tbody>
<tr>
<td>10L1</td>
<td>7</td>
<td>5</td>
<td>5</td>
<td>5 5 5 4 4 2 2 2 2 2</td>
</tr>
<tr>
<td>R</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10L1</td>
<td>7</td>
<td>5</td>
<td>7</td>
<td>6 6 6 4 3 3 3 3 3 3</td>
</tr>
<tr>
<td>C</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10L2</td>
<td>7</td>
<td>5</td>
<td>6</td>
<td>2 0 0 0 2 4 5 5 5 5</td>
</tr>
<tr>
<td>R</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10L2</td>
<td>7</td>
<td>5</td>
<td>5</td>
<td>2 2 0 0 3 5 5 5 5 5</td>
</tr>
<tr>
<td>C</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12L1</td>
<td>8</td>
<td>10</td>
<td>5</td>
<td>4 5 2 2 6 6 7 7 7 7</td>
</tr>
<tr>
<td>R</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12L1</td>
<td>8</td>
<td>10</td>
<td>6</td>
<td>6 4 5 2 8 8 7 8 8 8</td>
</tr>
<tr>
<td>C</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12L2</td>
<td>8</td>
<td>10</td>
<td>4</td>
<td>3 1 0 0 4 9 10 10 10</td>
</tr>
<tr>
<td>R</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12L2</td>
<td>8</td>
<td>10</td>
<td>5</td>
<td>2 2 1 0 6 10 10 10 10</td>
</tr>
<tr>
<td>C</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

3.6 Conclusion

A selective container routing and scheduling problem was proposed that extended the concept of resource sharing within a trucking company addressed in Chapter 2 to include multiple companies. The problem was defined as a variant of SVRP, which allows empty container pickup tasks to be selected along with those from outside carriers. Hence, the mathematical model was formulated from the perspective of a single private carrier. With a given cost structure for outsourcing and accepting the associated tasks, a better solution is expected to be found for a private carrier. This problem can be supported by the web-based task exchanging systems that have been developed in the field of collaborative logistics. In
order to help the proposed system and optimization problem, a pre-selection algorithm was proposed that allows a decision maker to solve problems of a reasonable size.

Through the performance comparison, the proposed solution approach provided a good quality of solutions with relatively short computation times. In addition, a series of numerical experiments was conducted to show that the proposed approach can help to reduce the operation cost, especially total travel time and operation time, and to reveal how the solution can change according to the setting of the instances (i.e., cost structure, the number of size of tasks from outside carriers, and location of the container yard). The results showed that the cost structure and associated selection of tasks had an influence on the solution. Thus, the problem needs to be solved with the exchanging costs and weights for the objective obtained from real-world data, which might provide a better insight about the benefit of the proposed solution to a decision maker.
CHAPTER 4 MULTI-DAY ROUTING AND SCHEDULING PROBLEM

4.1 Introduction

A plan of routing and scheduling for trucks moving a set of containers within a single day has been explored. For container movements, delivery of a container followed by pickup of the same container occur at the same location after a certain interval of time required for processing the container. Sometimes, given a limitation on the operation time of the trucking company, the following pickup tasks could not be finished within a planned interval because of relatively long process times for packing or unpacking containers. In terms of mathematical problems, this can generate an infeasible solution. Even if these tasks are served within the day, this can cause considerable increases in travel distance as well as operation time. Therefore, a trucking company could relocate some pickup tasks to the day after the corresponding delivery tasks were served (Veenstra, 2005).

The problems described in the previous chapters embedded these pickup tasks, which came from delivery tasks on the previous day, by maintaining independent empty and loaded container pickups. In order to shorten the operation time caused by the process, decoupling operation was incorporated. However, results showed that this operation could increase the total travel time at some level. In the previous experiments, care was taken to select time windows for customers with the following pickup tasks that would not generate an infeasible solution in the proposed problem. Nevertheless, there still was a chance that routes were infeasible even though they are very close to or better than the optimal solution, because the time window for pickup tasks depends on the preceding delivery task. If these
tasks can be postponed to the next day and assigned to a route with tasks on the day, this can help to reduce total operating cost and avoid the infeasibility of solutions. Additionally, in terms of street turning operations, postponing empty container pickup or delivery tasks could help to match more pair of tasks involved in street turnings, which can also reduce the costs.

The objective of this chapter is to propose a container truck routing and scheduling problem that relaxes the limitation on an operational period, which is defined as a multiday container truck routing and scheduling problem. Compared with the problem in Chapter 2, there are tasks on different days, and they can be posted to the next day and served with tasks planned to be served on that day. In general, since schedules for container movements (e.g., empty container movements or following container movements) can be flexible within a couple of days (Veenstra, 2005), this problem can be beneficial in reducing operating costs, in particular in the container truck industry. In this problem, we assume that a set of customers and associated time windows for a given short time period are known before planning truck routing and scheduling. For example, when a problem considers planning over two days, customers to be served on day 1 and those on day 2 and their time windows are given in advance. When the problem is solved, several customers can be postponed to day 2 and they are assigned to a route with customers planned on day 2. This might happen when customers cannot satisfy time constraints because of considerable intervals required for packing and unpacking containers, in which case such a postponement can reduce the total operating costs.
4.2 Literature review

The proposed problem is closely related to the Period Traveling Salesman Problem (PTSP) (e.g., Chao et al., 1995) and the Period Vehicle Routing Problem (PVRP) (e.g., Francis et al., 2006). This problem is a variant of TSP and VRP where routes are planned over multiple days. Compared with TSP and VRP, within a given period customers have a requirement of one or multiple visits, and a vehicle should complete all requested tasks for each customer. Usually, each customer has a service frequency to be visited, but does not restrict the day to be served. Thus, the days of visits are relaxed in this problem. The problem aims to find when each customer is visited and how a set of routes is planned to serve all customers for each day simultaneously in a way to minimize the total costs over multiple days. The main difference between PTSP and PVRP is that PTSP does not consider the capacity of vehicles and the quantity of a customer’s request.

Efforts to develop heuristic algorithms have been conducted to solve the PTSP and PVRP, because of the complexity of this class of problem. Among earlier works, Chao et al. (1995) developed a heuristic approach for the PTSP that is composed of generating an initial solution and then improving it with a record-to-record approach by chaining assignment patterns for each customer and adopting the 2-opt procedure to reduce the total distance. Paletta (2002) also proposed a simple tour construction based on an insertion approach with an embedded improvement procedure for the PTSP. Francis et al. (2006) provided a comprehensive literature survey of a class of PVRP and applied meta-heuristic approaches. Cordeau et al. (1997) developed a tabu search heuristic that was able to be applied to the PTSP, PVRP, and multi-depot vehicle routing problems. By considering time windows in the
PVRP, Cordeau et al. (2001) modified the tabu search algorithm to satisfy the newly added time constraints. Hemmelmayr et al. (2009) proposed a VNS approach for the PTSP and PVRP and showed that their algorithm can be competitive or perform better when compared with the existing approaches.

The multiday container truck routing and scheduling problem proposed in this chapter shares the main concept of PTSP and PVRP, which is that customers’ requests can be served within multiple days. However, in contrast with the PTSP and PVRP, each customer has a single request (i.e., service frequency is one for all customer), so that he is visited exactly once during the planned period. The proposed problem assumes that although each task requested by a customer has a given scheduled day, it can be postponed to the next day if a customer allows it to be postponed. This idea of postponing tasks in VRP has been studied in the dynamic multi-period vehicle routing problem (Angelelli et al., 2009; Wen et al., 2010). They considered the problem as a variant of PVRP without the service frequency of customer. In their problem, a dynamic component was incorporated in handling the tasks requested by a customer, which means that some tasks may arrive over time. In order to handle this dynamic problem, Angelelli et al. (2009) proposed short term routing strategies and Wen et al. (2010) developed a three-phase rolling horizon heuristic. While these approaches handle newly arrived requests during the time periods that are able to be postponed, our problem focuses on postponing tasks that are known in advance to the next period that is beneficial to move containers using the two operational strategies. In a similar fashion to our approach, Archetti et al. (2015) proposed a multi-period vehicle routing problem with due dates in which customers have to be visited between a release and a due
date, with a penalty cost for visits over the due date. They considered the problem with given
deterministic requests. However, while their problem had a broad range of time durations
in which a customer’s tasks could be completed and did not consider postponing the release
of a task to the next day, in the proposed problem tasks are originally planned to be served
within a day, but are able to be postponed. This assumption is more suitable for the
container routing problem. In addition, for each day, each customer has a given time window.

Even though truck operations for moving containers have some level of flexibility in
choosing service day, the problem considering multiday operations has been not
incorporated in this area. To our knowledge, the multiday truck container routing and
scheduling problem is first introduced here. The mathematical formulation is proposed
based on the problem including street turning and decoupling operations. By adopting
multiday operation, the benefit of these two operational strategies can be improved.

4.3 Modeling the problem

This section describes the multiday container truck routing and scheduling problem,
which is developed based on the problem addressed in Chapter 2. Thus, a trucking company
plans its schedule with consideration for street turning and decoupling operation strategies.
Additionally, assuming that moving containers can be scheduled over multiple days,
customers allow their tasks to be postponed from the day when they were originally
scheduled to the next time period. This assumption is justifiable in the container trucking
industry, because it is observed that some pickup tasks preceded by delivery tasks can be
served the day after delivery tasks are performed (Veenstra, 2005), or the time windows of
container shipments can range over several days in a real-life container problem based on a large port (Bai et al., 2015).

### 4.3.1 Problem description

The problem is defined with a planning horizon that consists of multiple days \( d \in D \). Suppose that all tasks requested by customers within the planning horizon are known in advance and no more additional requests are received within that period. With this assumption, the planning horizon can be set as a very short period (e.g., two or three days).

The problem is defined on a general graph \( G(N, A) \), where \( N = \{0,1, \ldots, n, n+1\} \) is the set of nodes and \( A = \{(i,j): i, j \in N\} \) is the set of arcs. In the set of nodes, there is one depot \( D = \{0, n+1\} \), one intermodal terminal \( T \), one container yard \( Y \), and six types of customers (i.e., \( C_{fd}, C_{fp}, C_{ed}, C_{ep}, C_{fped}, C_{epfd} \in C \)). Each customer has an original scheduled day to be visited \( z_i \) and should be visited within the given period. Even though customers can be visited on a later day through a postponement, each customer has a time window \([a_i, b_i]\) on the given day. Service time \( s_i \) and container process time (i.e., packing and unpacking time) \( p_i \) for customer \( i \) is given. In this problem, we assume that each customer has the same time window on different days (e.g., \([a_i, b_i]^{d=1} = [a_i, b_i]^{d=2}\)). However, the following pickup task after the preceding delivery task uses the time window for its preceding delivery task if these two tasks are performed on different days. For example, when a customer has a loaded container delivery task followed by an empty container pickup task, the time window for the empty container task is determined as the service start time plus container process time if it is served on the same day that the loaded container delivery task is served. On the other
hand, if the empty container pickup task is postponed to the next day, the time window for the loaded container delivery task is assigned to the time window for empty pickups.

The modified network designed in Chapter 2 is used for the multiday problem in the same fashion, which allows street turning and decoupling operations over the time period. The problem is designed to find an optimal set of routes and schedules that minimizes the total operating costs for a given planning horizon by allowing tasks to be served with postponements. Figure 4-1 represents examples of routes that can be generated by solving the proposed problem, compared with these obtained by the base model proposed in Chapter 2 for each day independently. In the general problem, a route is infeasible because a pickup task preceded by a delivery task violates time duration constraints, and no street turning operation is applied because the route could not find an appropriate customer for the operation (see Figure 4-1a). If the same set of customers is solved by the proposed approach (see Figure 4-1b), the infeasible customer can be assigned to a route on the next day along with several postponed customers, which makes all routes feasible. In addition, by postponing an empty container delivery task to the next day, the problem can find an allowable street turning arc, which helps to reduce total travel time as well as total operation time.
4.3.2 Mathematical formulation

The proposed problem is formulated as the am-TSPTW by allowing the postponement of tasks. The objective function aims to minimize the operating costs, which are composed of the number of trucks used, total travel distance traveled, total truck operation time, and the number of postponed tasks with their respective weights. By including the number of postponed tasks, unnecessary postponement of tasks will be prevented. A concept similar to this term was defined as inventory holding costs in the formulation proposed by Archetti et al. (2015). However, although our problem assumes that significant inventory cost is not...
incurred when a task is postponed, because container shipments can be planned within multiple days, the postponements are controlled by the weight factor.

\[
\begin{align*}
m\in Z &= \alpha_1 \sum_d \sum_v \Sigma_i x_{ij}^{vd} + \alpha_2 \sum_d \sum_v \Sigma_i x_{ij}^{vd} \\
&+ \alpha_3 \sum_d \sum_v T_{n+1}^{vd} - T_0^{vd} + \alpha_4 \sum_d (d - z_i) y_i^d
\end{align*}
\]

(4.1)

Subject to

\[
\Sigma_v \Sigma_i x_{ij} = y_i^d, \forall i \in C, \forall d \in D
\]

(4.2)

\[
\Sigma_i y_i^d = 1, \forall i \in C
\]

(4.3)

\[
(d - z_i)y_i^d \geq 0, \forall i \in C, \forall d \in D
\]

(4.4)

\[
\Sigma_j x_{ij}^{vd} - \Sigma_j x_{ij}^{vd} = 0, \forall i \in C, \forall v \in V, \forall d \in D
\]

(4.5)

\[
\Sigma_j x_{ij}^{vd} - \Sigma_j x_{ij}^{vd} = 0, \forall v \in V, \forall d \in D
\]

(4.6)

\[
\Sigma_j x_{ij}^{vd} \leq 1, \forall v \in V, \forall d \in D
\]

(4.7)

\[
T_{i}^{vd} + S_{i} + t_{ij} - T_{j}^{vd} \leq M(1 - x_{ij}^{vd}), \forall i, j \in N, \forall v \in V, \forall d \in D
\]

(4.8)

\[
T_{i}^{vd} + S_{i} + P_{i}(1 - x_{ij}^{vd}) + t_{ij} x_{ij}^{vd} \leq T_{j}^{vd} + M(1 - y_i^d), \forall (i, j) \in A, \forall v \in V, \forall d \in D
\]

(4.9)

\[
(b_{n+1} - (T_{i}^{vd} + S_{i} + t_{j,n+1}))y_i^d \geq 0, \forall (i, j) \in A, \forall v \in V, \forall d \in D
\]

(4.10)

\[
a_{j}(y_j^d - y_i^d) \leq T_{j}^{vd}, \forall (i, j) \in A, \forall v \in V, \forall d \in D
\]

(4.11)

\[
T_{j}^{vd} \leq b_{j}(y_j^d - y_i^d) + y_i^d b_{j}, \forall (i, j) \in A, \forall v \in V, \forall d \in D
\]

(4.12)

\[
y_j^d \leq \sum_{d'=1}^{d} y_i^{d'}, \forall (i, j) \in A, \forall v \in V, \forall d \in D
\]

(4.13)

\[
a_i y_i^d \leq T_{i}^{vd}, \forall i \in N, \forall v \in V, \forall d \in D
\]

(4.14)

\[
T_{i}^{vd} \leq b_{i}y_i^d, \forall i \in N, \forall v \in V, \forall d \in D
\]

(4.15)

\[
T_{n+1}^{vd} - T_{0}^{vd} \leq W, \forall v \in V, \forall d \in D
\]

(4.16)
, where

\( x_{ij}^{vd} \): 1 if vehicle \( v \) travels from node \( i \) and node \( j \) on day \( d \), otherwise 0

\( y_{id}^{d} \): 1 if task \( i \) is served on day \( d \), otherwise 0

\( c_{ij} \): Travel distance from node \( i \) and node \( j \)

\( t_{ij} \): Travel time from node \( i \) and node \( j \)

\( z_{i} \): A given day when task \( i \) is expected to be completed

\( V \): Set of vehicles

\( D \): Set of depot, \( \{0, n + 1\} \)

\( C \): Set of customers \( \{1, 2, ..., n\} \)

\( N \): Set of nodes, \( D \cup C \)

\( A_{C} \): Set of pairs of two consecutive tasks

\( T_{i}^{vd} \): Service start time at node \( i \) on day \( d \) by vehicle \( v \)

\( S_{i} \): Duration for service at node \( i \)

\( P_{i} \): Duration for packing or unpacking of container at node \( i \)

\( a_{i}, b_{i} \): Lower and upper time window at node \( i \)

\( W \): Maximum work-shift hours for driver

\( \alpha_{1}, \alpha_{2}, \alpha_{3}, \alpha_{4} \): Weighted factors

\( M \): Arbitrary large number

Constraints (4.2) and (4.3) ensure that every customer is visited exactly once in all routes for the entire period. Constraints (4.4) guarantee that each customer can be postponed, but cannot be advanced. Constraints (4.5) and (4.6) enforce flow conservation
for each tour and for a depot, respectively. Constraints (4.7) ensure that vehicles can be used for each route at most once. Constraints (4.8) calculate the service start time at a later customer, which is the sum of the service start time at the current customer, service time at that customer, and travel time between the two customers. Constraints (4.9) represent the beginning of the time window for the consecutive tasks, $C_{f_{ped}}$ and $C_{ep_{fd}}$, which is calculated as the sum of the service start time of the corresponding $C_{ed}$ or $C_{fd}$, service time, and container process time. If the consecutive task is served with a waiting operation, it saves service time for dropping or picking up a container. On the other hand, if the decoupling occurs at that location, dropping or picking up time should be included to compute the time window dependent on the preceding task. Constraints (4.10) ensure that a customer preceded by a given task should be postponed to the next day if this task cannot be completed on the scheduled day. Constraints (4.11) and (4.12) ensure that if two consecutive tasks in $A_c$ are served within the same day, the time window for the following task is determined by Constraints (4.10) and the predetermined end of the time window, otherwise the following task uses the same time window for the preceding task. Constraints (4.13) guarantee that for two consecutive tasks in $A_c$, the following task cannot be assigned to the day before the preceding task is finished. Constraints (4.14) and (4.15) provide time windows for each customer. Constraints (4.16) enforce drivers’ maximum work-shift hours for each tour.

4.4 Solution approach

A VNS algorithm has been used to solve a class of VRPs with a respectable performance. Also, several studies have applied a VNS to solve the PVRP (Hemmelmayr,
Doerner and Hartl, 2009; Tricoire et al., 2010). Since the proposed problem can be seen as a variant of the PVRP, the algorithm based on a VNS can be applied to solve the problem within a reasonable computational time. Several modifications on the algorithm proposed in Chapter 2 could be considered for extension to multiday consideration with postponement.

4.4.1 Feasible arcs

In order to reduce the search space and apply the proposed algorithm with small modifications, a set of feasible arcs is determined in advance. A simple rule based on time constraints and the originally scheduled day is applied to create the list as follows.

- Rule 1: A task that is scheduled on day $d + 1$ cannot be connected to a task that is scheduled on day $d$, which prevents the task from being advanced.
- Rule 2: The arc $(i, j)$ cannot exist if the sum of the earliest service start time at node $i$, service time at node $i$, and travel time between two nodes $i$ and $j$ exceeds the latest service start time at node $j$.

This set of feasible arcs is used not only for exploring neighborhoods of the current solution, but also in constructing the initial solutions in the proposed meta-heuristic algorithm.

4.4.2 Modifications to the two-stage algorithm

Among the eight neighborhood structures described in Chapter 2, six neighborhood structures are used in this algorithm: Intra-route insertion $N_1$, Inter-route insertion $N_2$,
Intra-route swap $N_3$, Inter-route swap $N_4$, Intra-route 2-opt $N_5$, and Inter-route 2-opt $N_6$. Additionally, four more neighborhood structures are developed in order to generate further perturbed neighborhoods effectively across different days as follows.

- **Inter-route insertion with day assignment $N_7$** (Figure 3-3a): A customer is randomly selected from its current location on a randomly selected route and inserted to a randomly selected location on a different route. After that, a new randomly selected day is assigned to each route.

- **Inter-route 2-opt with day assignment $N_8$** (Figure 3-3b): Two arcs are selected and removed from the same route. Two preceding customers of each arc and two following customers of each arc are reconnected; thus, the order between the two arcs is reversed. After this step, a new randomly selected day is assigned to each route.

- **Inter-route swap with day assignment $N_9$** (Figure 3-3c): Two customers are randomly selected from two different routes and their positions are switched with each other. After this, a new randomly selected day is assigned to each route.

- **Cyclic inter-route double insertion $N_{10}$** (Figure 3-3d): Two consecutive customers are randomly selected from their current location on the route, and each customer reinserted to a respective random location on two different routes randomly selected.

- **Cyclic inter-route insertion $N_{11}$** (Figure 3-3e): A customer is randomly selected from its current location on a randomly selected route and inserted to a location of a different route randomly selected. Then, the customer that is at the inserted location is moved out and inserted to a location on a route on a different day.
With these additional neighborhood structures, a deterministic order of neighborhood structures is used for GVNS at the master level, which is \( N_k = \{N_7, N_8, N_9, N_{10}, N_{11}\} \). In the local search, a VND algorithm applies randomization of the set of neighborhood structures with all structures.

\[
N_l \leftarrow \text{random sample} \left(\{N_2, N_4, N_5, N_6\}, 1\right) + \text{random sample (remainings)},
\]

Figure 4-2 Additional neighborhood structures
where $\text{random sample}(A,n)$ is a function that randomly generates $n$ elements of A. The first element of neighborhood is restricted by \(\{N_2, N_4, N_5, N_6\}\) at each iteration.

### 4.5 Numerical experiments

#### 4.5.2 Modified test instances

For numerical experiments in the multiday environment, several modifications were conducted based on the 60 test instances generated in Chapter 2. Assuming a planning horizon of two days, two sets of test instances were designed to test the proposed problem. In the first set, all tasks are scheduled on day 1 and no task is planned on day 2. On the other hand, the second set has tasks scheduled over a two-day period; 70% of customers have a task released on day 1 and the remainder are on day 2. In order to relax the restriction on time windows for the consecutive tasks and to allow for long process times in packing and unpacking the containers, time windows for each customer are randomly generated according to the following criteria.

- Time window for the depot and container yard on each day: 6 am - 12 am
- The beginning of time window for $C_{fd}, C_{ed}, C_{fp}, C_{ep}$ on day 1: a uniform random variable between 8 am and 6 pm
- The beginning of time window for $C_{fp}, C_{ep}$ on day 2: a uniform random variable between 8 am and 6 pm
• The beginning of time window for $C_{fd}, C_{ed}$ on day 2: a uniform random variable between 8 am and 12 pm, which prevents infeasibility of the following pickup tasks on day 2.

• The time window interval for customer $C_{fd}, C_{fp}, C_{ed}, C_{ep}$: a uniform random variable from 60 minutes to 240 minutes, in 30-minute increments

• The end of time window for $C_{ped}, C_{eped}$: 12 am

• The process time for customer $C_{fd}, C_{ed}$: a uniform random variable from 60 minutes to 300 minutes, in 60-minute increments.

4.5.2 Computational results

Evaluation of performance

The mathematical model was solved by the commercial solver on a Mac-PC having a 2.5 GHz i5 processor and 16 GB of RAM. The two-stage algorithm based on the GVNS was run with the set of neighborhood structures described in the previous section. The maximum number of iterations of 30 and the maximum number of non-improvements of 10 were set as the stopping criteria. For the acceptance threshold of solutions at master level, the initial temperature parameter $T$ is set to be 100 and is decreased by 10% at every iteration. The weight factors of the objective were assumed to be $\alpha_1 = 100, \alpha_2 = 1, \alpha_3 = 35/60, \alpha_4 = 1$. With the weights for postponements, the operation easily allows tasks to be transferred to the next day.
The performance of the proposed algorithm based on GVNS were compared with that of the commercial solver as shown in Table 4-1 to Table 4-4. Results show that the proposed approach can find an optimal solution or near optimal solution within a reasonable computational time. The algorithm performed better even in several test instances of small size problems. In large size instances, while the solver was not able to find the solution for test instances of over 40 customers within an hour, the algorithm found the solution. In addition, the proposed problem performed well under two different environments; one has only first day tasks that are able to be postponed to the next day, and the other has tasks over a two-day period, where tasks released on day 1 are allowed to suspend to day 2.

Table 4-1. Comparison of GVNS against Solver for the test instances having all tasks released on day 1 (small size instances)

<table>
<thead>
<tr>
<th>Instances</th>
<th>Solver(Gurobi)</th>
<th>GVNS</th>
</tr>
</thead>
<tbody>
<tr>
<td>01L1R</td>
<td>625.60</td>
<td>0.28</td>
</tr>
<tr>
<td>01L1C</td>
<td>652.88</td>
<td>0.06</td>
</tr>
<tr>
<td>01L2R</td>
<td>458.09</td>
<td>0.22</td>
</tr>
<tr>
<td>01L2C</td>
<td>475.68</td>
<td>0.18</td>
</tr>
<tr>
<td>02L1R</td>
<td>867.43</td>
<td>0.45</td>
</tr>
<tr>
<td>02L1C</td>
<td>891.51</td>
<td>0.36</td>
</tr>
<tr>
<td>02L2R</td>
<td>688.67</td>
<td>0.52</td>
</tr>
<tr>
<td>02L2C</td>
<td>700.97</td>
<td>0.30</td>
</tr>
<tr>
<td>03L1R</td>
<td>682.52</td>
<td>0.82</td>
</tr>
<tr>
<td>03L1C</td>
<td>574.54</td>
<td>0.68</td>
</tr>
<tr>
<td>03L2R</td>
<td>610.53</td>
<td>0.74</td>
</tr>
<tr>
<td>03L2C</td>
<td>512.08</td>
<td>0.75</td>
</tr>
<tr>
<td>04L1R</td>
<td>1,216.82</td>
<td>7.40</td>
</tr>
<tr>
<td>04L1C</td>
<td>1,272.47</td>
<td>2.62</td>
</tr>
<tr>
<td>04L2R</td>
<td>1,089.89</td>
<td>44.78</td>
</tr>
<tr>
<td>04L2C</td>
<td>1,118.83</td>
<td>8.45</td>
</tr>
<tr>
<td>Instances</td>
<td>Solver (Gurobi)</td>
<td>GVNS</td>
</tr>
<tr>
<td>-----------</td>
<td>----------------</td>
<td>------</td>
</tr>
<tr>
<td></td>
<td>Obj. Value</td>
<td>Time (s)</td>
</tr>
<tr>
<td>05L1R</td>
<td>1,399.39</td>
<td>3,386.74</td>
</tr>
<tr>
<td>05L1C</td>
<td>1,632.24</td>
<td>3,600.00</td>
</tr>
<tr>
<td>05L2R</td>
<td>1,037.77</td>
<td>3,600.01</td>
</tr>
<tr>
<td>05L2C</td>
<td>1,289.35</td>
<td>3,600.01</td>
</tr>
<tr>
<td>06L1R</td>
<td>1,428.28</td>
<td>3,600.01</td>
</tr>
<tr>
<td>06L1C</td>
<td>1,660.04</td>
<td>3,600.01</td>
</tr>
<tr>
<td>06L2R</td>
<td>1,017.23</td>
<td>3,600.01</td>
</tr>
<tr>
<td>06L2C</td>
<td>1,202.85</td>
<td>3,600.01</td>
</tr>
<tr>
<td>07L1R</td>
<td>1,887.69</td>
<td>3,600.01</td>
</tr>
<tr>
<td>07L1C</td>
<td>1,632.24</td>
<td>3,600.00</td>
</tr>
<tr>
<td>07L2R</td>
<td>1,263.35</td>
<td>3,600.01</td>
</tr>
<tr>
<td>07L2C</td>
<td>1,354.07</td>
<td>3,600.01</td>
</tr>
<tr>
<td>08L1R</td>
<td>1,564.57</td>
<td>3,600.01</td>
</tr>
<tr>
<td>08L1C</td>
<td>1,968.08</td>
<td>3,600.01</td>
</tr>
<tr>
<td>08L2R</td>
<td>1,328.64</td>
<td>3,600.01</td>
</tr>
<tr>
<td>08L2C</td>
<td>1,611.37</td>
<td>3,600.02</td>
</tr>
<tr>
<td>09L1R</td>
<td>2,592.55</td>
<td>3,600.01</td>
</tr>
<tr>
<td>09L1C</td>
<td>2,858.62</td>
<td>3,600.01</td>
</tr>
<tr>
<td>09L2R</td>
<td>2,206.89</td>
<td>3,600.01</td>
</tr>
<tr>
<td>09L2C</td>
<td>2,410.37</td>
<td>3,600.02</td>
</tr>
</tbody>
</table>

<sup>a</sup> Gap = (Objective bound – Current Incumbent Solution) / Current Incumbent Solution (%)

<sup>b</sup> Gap = (Objective form the solver – Best objective from GVNS) / Best objective from GVNS (%)

Table 4-2 Comparison of GVNS against Solver for the test instances having all tasks released on day 1 (large size instances)
<table>
<thead>
<tr>
<th>Instances</th>
<th>Solver(Gurobi)</th>
<th>GVNS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Obj. Value</td>
<td>Time (s)</td>
</tr>
<tr>
<td>01L1R</td>
<td>628.60</td>
<td>0.06</td>
</tr>
<tr>
<td>01L1C</td>
<td>655.88</td>
<td>0.07</td>
</tr>
<tr>
<td>01L2R</td>
<td>461.09</td>
<td>0.06</td>
</tr>
<tr>
<td>01L2C</td>
<td>478.68</td>
<td>0.05</td>
</tr>
<tr>
<td>02L1R</td>
<td>870.43</td>
<td>0.32</td>
</tr>
<tr>
<td>02L1C</td>
<td>894.51</td>
<td>0.24</td>
</tr>
<tr>
<td>02L2R</td>
<td>691.67</td>
<td>0.25</td>
</tr>
<tr>
<td>02L2C</td>
<td>703.97</td>
<td>0.24</td>
</tr>
<tr>
<td>03L1R</td>
<td>693.20</td>
<td>0.29</td>
</tr>
<tr>
<td>03L1C</td>
<td>576.54</td>
<td>0.28</td>
</tr>
<tr>
<td>03L2R</td>
<td>612.53</td>
<td>0.38</td>
</tr>
<tr>
<td>03L2C</td>
<td>514.08</td>
<td>0.30</td>
</tr>
<tr>
<td>04L1R</td>
<td>1,317.35</td>
<td>1.16</td>
</tr>
<tr>
<td>04L1C</td>
<td>1,381.80</td>
<td>1.01</td>
</tr>
<tr>
<td>04L2R</td>
<td>1,123.10</td>
<td>5.10</td>
</tr>
<tr>
<td>04L2C</td>
<td>1,114.83</td>
<td>3.69</td>
</tr>
<tr>
<td>05L1R</td>
<td>1,476.09</td>
<td>884.51</td>
</tr>
<tr>
<td>05L1C</td>
<td>1,632.24</td>
<td>1,088.74</td>
</tr>
</tbody>
</table>

\(^a\) Gap = (Objective bound – Current Incumbent Solution) / Current Incumbent Solution (%)
\(^b\) Gap = (Objective form the solver – Best objective from GVNS) / Best objective from GVNS (%)

Table 4-3. Comparison of GVNS against Solver for the test instances having tasks over a two-day period (small size instances)
Table 4-4 Comparison of GVNS against Solver for the test instances having tasks over a two-day period (large size instances)

<table>
<thead>
<tr>
<th>Instances</th>
<th>Solver(Gurobi)</th>
<th>GVNS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Obj. Value</td>
<td>Time (s)</td>
</tr>
<tr>
<td>10L1R</td>
<td>3,065.08</td>
<td>3,600.01</td>
</tr>
<tr>
<td>10L1C</td>
<td>3,572.02</td>
<td>3,600.02</td>
</tr>
<tr>
<td>10L2R</td>
<td>2,348.25</td>
<td>3,600.01</td>
</tr>
<tr>
<td>10L2C</td>
<td>2,561.98</td>
<td>3,600.02</td>
</tr>
<tr>
<td>11L1R</td>
<td>3,476.94</td>
<td>3,600.03</td>
</tr>
<tr>
<td>11L1C</td>
<td>4,032.37</td>
<td>3,600.03</td>
</tr>
<tr>
<td>11L2R</td>
<td>2,818.60</td>
<td>3,600.03</td>
</tr>
<tr>
<td>11L2C</td>
<td>3,013.33</td>
<td>3,600.03</td>
</tr>
<tr>
<td>12L1R</td>
<td>5,408.98</td>
<td>3,600.37</td>
</tr>
<tr>
<td>12L1C</td>
<td>5,990.97</td>
<td>3,600.11</td>
</tr>
<tr>
<td>12L2R</td>
<td>3,941.98</td>
<td>3,600.04</td>
</tr>
<tr>
<td>12L2C</td>
<td>4,545.68</td>
<td>3,600.03</td>
</tr>
<tr>
<td>13L1R</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

\(^a\)Gap = (Objective bound – Current Incumbent Solution) / Current Incumbent Solution (%)

\(^b\)Gap = (Objective from the solver – Best objective from GVNS) / Best objective from GVNS (%)
Table 4-5 Summary of performance for the test instances having all tasks released on day 1

<table>
<thead>
<tr>
<th>Set of instances (# instances)</th>
<th>Avg. Gap of Objective Value</th>
<th>Avg. Run Time (sec.)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>GVNS vs Solver</td>
<td>Solver</td>
</tr>
<tr>
<td>1-5(20)</td>
<td>0.00%</td>
<td>712.77</td>
</tr>
<tr>
<td>6-9(16)</td>
<td>0.66%</td>
<td>3,600.01</td>
</tr>
<tr>
<td>10 (4)</td>
<td>1.59%</td>
<td>3,600.01</td>
</tr>
<tr>
<td>11 (4)</td>
<td>4.55%</td>
<td>3,600.01</td>
</tr>
<tr>
<td>12 (4)</td>
<td>-</td>
<td>3,600.01</td>
</tr>
<tr>
<td>13 (4)</td>
<td>-</td>
<td>3,600.01</td>
</tr>
<tr>
<td>14 (4)</td>
<td>-</td>
<td>3,600.01</td>
</tr>
<tr>
<td>15 (4)</td>
<td>-</td>
<td>3,600.01</td>
</tr>
</tbody>
</table>

Table 4-6 Summary of performance for the test instances having all tasks released over a two-day period

<table>
<thead>
<tr>
<th>Set of instances (# instances)</th>
<th>Avg. Gap of Objective Value</th>
<th>Avg. Run Time (sec.)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>GVNS vs Solver</td>
<td>Solver</td>
</tr>
<tr>
<td>1-5(20)</td>
<td>0.00%</td>
<td>169.13</td>
</tr>
<tr>
<td>6-9(16)</td>
<td>0.50%</td>
<td>3,600.01</td>
</tr>
<tr>
<td>10 (4)</td>
<td>0.38%</td>
<td>3,600.01</td>
</tr>
<tr>
<td>11 (4)</td>
<td>0.53%</td>
<td>3,600.01</td>
</tr>
<tr>
<td>12 (4)</td>
<td>-</td>
<td>3,600.01</td>
</tr>
<tr>
<td>13 (4)</td>
<td>-</td>
<td>3,600.01</td>
</tr>
<tr>
<td>14 (4)</td>
<td>-</td>
<td>3,600.01</td>
</tr>
<tr>
<td>15 (4)</td>
<td>-</td>
<td>3,600.01</td>
</tr>
</tbody>
</table>
Evaluation of the weight factor controlling postponed tasks

The solutions were tested by changing the weights of the number of postponed tasks. As these weights increase, the postponement of tasks might be restricted so that only a limited number of tasks can be suspended. If decision makers or customers are reluctant to postpone their tasks and share their resources over different days, only the tasks that might violate the operation duration on a scheduled day will be postponed to the next day. In order to evaluate the effect of weightings on the solution, the problem was tested with different weights.

Figure 4-3 and Figure 4-4 show the number of vehicle used, total travel distance and total operating time obtained from instance set 10 and 12 according to the weights of postponed tasks, respectively. This experiment was conducted on the second instance set in which tasks were generated over a two-day period. The results show that travel distance and operation time decrease when postponements are relaxed. This means that the postponements were very effective in terms of reduction of travel time and operation time. Compared with the model considering the level of resource sharing on a single day (e.g., the base model in Chapter 2), which refers to the case of restricted postponements in this experiment, the routing plan with postponement representing resource sharing over the planning horizon is beneficial at some level. However, marginal effects were observed in the location set 2 compared to the location set 2. The reason for this might be that most of the
tasks were able to be served within a day because of the relatively short distance to the container yard.

**Figure 4-3** Effect of postponements controlled by the weight from instances 10
Figure 4-4 Effect of postponements controlled by the weight from instances 12
If the willingness to postpone tasks vary widely across customers, different weights according to the tasks can apply to the term of controlling postponements in the objective by chaining that into $\sum_d \sum_i \alpha_i^d (d - z_i) y_t^d$. For example, if a customer does not allow its task to be postponed because of urgent need of delivery, arbitrary high number can be assigned to its weight factor.

4.6 Conclusion

The single-period problem presented in Chapter 2 was extended to the problem considering multi-period operations in which tasks scheduled for a certain day can be postponed to following periods. We assumed that while customers have a fixed time window in their operation period as defined in the previous problem, they can at some level be completed within a relaxed service completion period. Therefore, tasks can be suspended and shared with the next day's resources, with such postponements being controlled by the weight factor of the objective. This assumption can be justifiable for drayage operations, because most of such tasks are not urgent and are planned over a period of days or weeks. This consideration can also be suitable for tasks requiring long container processing times or facing the possibility that the subsequent task cannot be completed within a day. Five neighborhood structures that generate a neighborhood solution by chaining the current solution across the different days were added into the GVNS algorithm. This problem and the modified GVNS were tested with randomly generated test instances with a two-day planning horizon. The results show that this solution approach can find good quality solutions within a reasonable computational time.
CHAPTER 5 CONCLUSION AND FINAL REMARKS

5.1 Summary of contribution

The main contribution of this dissertation is summarized as follows:

• Extended the literature on routing and scheduling problems that arise from container movements by adopting street turning and decoupling operations simultaneously

• Included temporal precedence constraints and site-dependent constraints

• Developed a feasibility check algorithm for temporal precedence constraints

• Proposed modified routing networks for site-dependent constraints

• Developed three different routing and scheduling problems of container truck in a shared resource environment
  • Base model
  • Selective routing and scheduling problem with selective empty container pickup
  • Multi-day routing and scheduling problem

• Developed optimization heuristics approaches (GVNS)

This dissertation proposed three different truck container routing and scheduling problems under a shared freight transportation environment. In the increasingly competitive trucking industry, resource sharing raises a solution approach to increase the benefits of individual carriers and ultimately improve the efficiency of the entire freight transportation system in an urban area. In addition, this approach can also provide good
strategic operational solutions that resolve the unproductive truck movements that inherently result from moving containers in a hinterland. To enable a trucking company to operate its fleet with shared resources, the proposed problem adopted two operational strategies—street turnings and decoupling operations—and temporal precedence constraints in addition to the time constraints usually included in the VRPTW. Thus, the proposed problems are expected to be more suitable for the practical application of truck operations under a shared transportation environment. Each problem is assumed to be applied to different levels of resource sharing as follows.

The problem in which trucks and containers assigned to customers within a carrier and within an operation period are shared with other customers within the same carrier and period was modeled in Chapter 2. In contrast with previous studies, the problem was designed to account for whether each customer allows each operation strategy or not through the network modification procedure, which can address practical and institutional barriers that have been pointed out for the implementation of these strategies. Through numerical experiments conducted on the randomly generated test instances in this study, the benefits of each strategy and the influences of the various objectives on the solution were evaluated.

The selective container truck routing and scheduling problem with collaborative resource sharing is proposed in Chapter 3. At this level of resource sharing, empty containers can be shared across the carriers participating in the coalition. Thus, the problem is formulated as a SVRP with time windows from a carrier perspective. Without loss of
generality, we assume that only empty container pickup tasks that are non-revenue are able to be exchanged between the private carrier and outside carriers, through a web-based information exchange system that has been widely utilized in shared transportation systems for passengers and even for freight. For this collaborative problem, a pre-selection algorithm that provides the best potential candidate set of tasks into the routing and scheduling problem is proposed and evaluated. By comparing the objective values obtained from this approach with those employing non-collaborative operations, we find that collaborative operation can reduce the total operation cost, even though the total cost includes the task exchange cost.

The single-period problem presented in Chapter 2 was extended to the problem considering multi-period operations, in which tasks scheduled on a certain day can be postponed to following periods. In this problem, customers have a fixed time window in their operation period, but a more relaxed service completion period at some level, which is a reasonable operational approach for the drayage industry, because most tasks are not urgent and are planned over a period of days or weeks. Thus, tasks can be suspended and use resources assigned to a different day, which will not increase the total cost much. In addition, this approach was designed to be suitable for tasks requiring long container processing times or that face the possibility that the subsequent task cannot be completed within a day. This problem was tested with a randomly generated test set in which the planning horizon was set to be two days, and the processing times and the ranges of the earliest service times were relaxed compared with the test instances used in Chapters 2 and 3.
In order to solve the routing and scheduling problem with temporal constraints, which are computationally more expensive, a solution approach based on the GVNS was proposed for all three problems. In this approach, a novel feasibility check algorithm was developed to deal with flexible time windows resulting from temporal precedence constraints. Several modifications in the solution approach were made to utilize suitable neighborhood search methods for each problem. The solution approach can easily be adopted to solve a class of VRPTWs considering temporal precedence constraints.

This dissertation mainly focused on modeling problems in the emerging shared freight transportation environment and developing algorithms to improve the efficiency of drayage operations from the perspective of individual carriers. The proposed problems showed the benefits of truck operations with shared resources. The capability for the algorithm to find both quickly and accurately an optimal or near-optimal truck operation plan for picking up and delivering containers in a hinterland was verified in each chapter. Consequently, this dissertation offers a decision-support tool to drayage companies when they are willing to apply some level of resource sharing into their operations.

5.2 Future research

This dissertation explored the routing and scheduling problems for moving containers by truck in a local area with different resource-sharing options. While this can provide an operational plan to carrier operators that want to improve the current inefficient movement of containers, further research remains to be conducted for improvements and
practical applications. The proposed problems assume that a vehicle can be used once. However, these constraints can be extended to that a vehicle can start its operation again after coming back to the depot without violating a driver’s work-shift hours. The problems can be modified based on the formulation proposed by (Recker, 1995).

In the proposed problems, heterogeneity of containers was ignored. The type of containers requested by customers can vary depending on their business. For example, while one customer wants a general container, another needs a refrigerated container. Such characteristics can be added into the problems as site-dependent constraints. In addition, in this study we assumed that trucks and containers were the resources that could be shared among customers, and that chassis suitable to containers were sufficiently supplied at any location. However, in the real world, chassis belong to different customers and sometimes only fit with a certain container (Genevieve et al., 2013). Thus, chassis need to be considered as an independent resource, and the problems can be extended to incorporated chassis movements as well. In this case, the problems combine container movements with chassis movements and consider the suitability of two resources at a customer location simultaneously.

Uncertainty could be added into the modeling of container truck movements. The proposed problems were designed to solve the problem in a deterministic way to show the capabilities of shared resource transportation in the container truck operation plan. However, there are two components in the container routing network, travel time uncertainty and service time uncertainty, which have been widely considered in a class of
VRPs. As previously mentioned, because container trucks travel through urban roads, they can encounter various unexpected events such as accidents and congestion, as urban freight trucks often do. In terms of service time, the processing of containers, which represents unpacking loaded containers or packing empty containers, is required at a customer location. Practically, the exact time taken to complete this process could not be determined. In order to consider these uncertainties, a stochastic approach can be applied to these problems, which is expected to provide more reliable routing and scheduling plans. In addition, if new tasks are received during operations, the current routing plan needs to be changed to accommodate these tasks. For this consideration, a dynamic approach can be adopted. In particular, the multiday routing problem can be extended to solve for this by repeatedly using a rolling horizon framework.

For resource sharing among multiple carriers, this dissertation focused on truck operations from a carrier’s point of view in Chapter 3. Here, the routes for outside carriers were ignored. However, their routes and schedules could affect the exchanging of their tasks with others. Thus, solving the problem including other carriers’ routes together might help to find better solutions for all participants. To extend the multi-carrier problem, the concept of the multi-depot problem can be introduced. Also, while this study utilized a simple cost structure for exchanging tasks, more sophisticated cost structures and cost allocation approaches between carriers need to be studied for the multi-carrier problem.

With the rapid development of automated vehicles, in the near future freight transportation will launch automated trucks. In drayage operations, autonomous trucks
have already been used to transport containers within a port area and will be used beyond ports eventually, although their current routes are simpler than those for urban distribution. In order to consider these trucks with conventional trucks, the routing problem can be extended into a heterogeneous routing problem to incorporate the characteristics of automated trucks into the constraints, including lifting the restrictions on drivers’ work hours. The proposed formulation and solution approach can be easily extended to solving the problem including autonomous truck operations, which will be briefly discussed in the following section.

5.3 Autonomous truck fleet routing and scheduling problems

This section discusses an extension of the proposed problem that considers drayage operations for a fleet having autonomous and conventional trucks at the same time. As technology for autonomous vehicles has significantly advanced and associated legislations has been enacted, the trucking industry is expecting to operate them in the near future. Uber’s autonomous truck already made the first real-world test that was to travel over 120 miles for delivering beers on October, 2016 (Fortune, 2016). The introduction of autonomous vehicles to drayage operations would bring some benefits such as reductions in labor cost, operation cost, and crashes. Despite of the benefit, autonomous trucks might gradually replace the existing conventional trucks in a trucking company due to the higher purchasing cost. Thus, the company is required to plan routes and schedules for a fleet of heterogeneous vehicles that is composed of autonomous vehicles and conventional vehicles with drivers.
In conventional truck operations, hours of service regulations apply to truck drivers, which means that a conventional truck should come back to the depot within the limit of driving and working time for the drivers. In contrast, autonomous trucks can be unrestrictedly operated during the daily operation period, so they are capable of serving more tasks than conventional ones. However, it is expected that they have a higher fixed cost for their operation than conventional vehicles because of their possession and maintenance cost. Therefore, determining an optimal routing plan for a fleet having those two types of vehicles can be defined as the heterogeneous vehicle routing problem (HVRP) (Koç et al., 2016). Since the mathematical formulation proposed in Chapter 2 which was designed to solve the homogeneous vehicle routing problem considered each route constructed by each vehicle, that can easily apply to the problem with autonomous vehicles.

Compared to the problem for a homogenous fleet, the objective has a fixed cost term addressing two types of fixed costs according to the vehicle types. Besides, constraints that enforce the limit of maximum driver’s working hours have different limits by vehicle type. This problem takes a fixed fleet into account and determines the composition of fleet as well as vehicle routes and schedules. Consequently, the problem is formulated as follows.

\[
\min Z = \alpha_1 \sum_\nu \sum_j x^\nu_{0j} + \alpha_2 \sum_\nu \sum_{ij} c^\nu_{ij} x^\nu_{ij} + \alpha_3 \sum_\nu T^\nu_{n+1} - T^\nu_0
\]

(5.1)

Subject to

(2.2) - (2.9)

\[
T^\nu_{n+1} - T^\nu_0 \leq W^\nu, \ \forall \ \nu \in V
\]

(5.2)
where a set of vehicles consist of conventional vehicles $V_1$ and autonomous vehicles $V_2$ (i.e., $\{V_1, V_2\} \in V$). The two-stage solution approach with GVNS algorithm is able to apply with a modification on the feasibility check procedure where different route durations are functioned according to the type of vehicles.

A potential use case was presented in Appendix A, by solving the test instances generated in Chapter 2. Results suggest that the proposed problem and associated algorithms are a promising approach for fleet operations with mixed autonomous and conventional vehicles.
APPENDIX 1. A HETEROGENEOUS FLEET OF AUTONOMOUS AND CONVENTIONAL TRUCKS

The purpose of this appendix is to show the potential use of the models proposed in this dissertation to find a routing plan with a mixed fleet having several autonomous trucks. The capability is evaluated with the base model developed in Chapter 2 on the same instances. From the results obtained from numerical experiments in Chapter 2, a fixed number of conventional trucks were determined in advance as shown in Table A-0-1. Simply, all test instances had one autonomous trucks. It is assumed that the fixed cost for autonomous trucks is twice that for conventional trucks (i.e., $\alpha^v_1 = 100, v \in$ conventional trucks and $\alpha^v_1 = 200, v \in$ autonomous trucks) ; the limit of operation duration for autonomous trucks is equal to daily fleet operation hours (i.e., 18 hours) while that for conventional trucks is driver’s work-shift hours that have used for the previous problems (i.e., 10 hours).

Table A-0-1 Number of customers and trucks for each test instance

<table>
<thead>
<tr>
<th>Customer set</th>
<th>$C_{fd}$</th>
<th>$C_{fp}$</th>
<th>$C_{ed}$</th>
<th>$C_{ep}$</th>
<th>$C_{epfd}$</th>
<th>$C_{fped}$</th>
<th>total</th>
<th>Conventional</th>
<th>Autonomous</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>5</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>6</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>8</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>10</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>12</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>14</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>4</td>
<td>3</td>
<td>16</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>9</td>
<td>5</td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>5</td>
<td>4</td>
<td>20</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>25</td>
<td>5</td>
<td>1</td>
</tr>
</tbody>
</table>
Table A-0-2 represent that the proposed solution approach was clearly performed well to solve the container truck routing and scheduling problem with a heterogeneous fleet including autonomous trucks in terms of computational time.

<table>
<thead>
<tr>
<th>Instances</th>
<th>Solver(Gurobi)</th>
<th>GVNS</th>
</tr>
</thead>
<tbody>
<tr>
<td>01L1R</td>
<td>515.24</td>
<td>0.01</td>
</tr>
<tr>
<td>01L1C</td>
<td>522.83</td>
<td>0.02</td>
</tr>
<tr>
<td>01L2R</td>
<td>515.24</td>
<td>0.01</td>
</tr>
<tr>
<td>01L2C</td>
<td>522.83</td>
<td>0.02</td>
</tr>
<tr>
<td>02L1R</td>
<td>745.62</td>
<td>0.05</td>
</tr>
<tr>
<td>02L1C</td>
<td>786.19</td>
<td>0.05</td>
</tr>
<tr>
<td>02L2R</td>
<td>731.48</td>
<td>0.07</td>
</tr>
<tr>
<td>02L2C</td>
<td>774.15</td>
<td>0.07</td>
</tr>
<tr>
<td>03L1R</td>
<td>745.94</td>
<td>0.07</td>
</tr>
<tr>
<td>03L1C</td>
<td>776.50</td>
<td>0.09</td>
</tr>
<tr>
<td>03L2R</td>
<td>773.49</td>
<td>0.10</td>
</tr>
<tr>
<td>03L2C</td>
<td>763.63</td>
<td>0.10</td>
</tr>
<tr>
<td>04L1R</td>
<td>935.83</td>
<td>0.18</td>
</tr>
<tr>
<td>04L1C</td>
<td>1,033.92</td>
<td>0.21</td>
</tr>
<tr>
<td>04L2R</td>
<td>773.60</td>
<td>0.26</td>
</tr>
<tr>
<td>04L2C</td>
<td>830.06</td>
<td>0.33</td>
</tr>
<tr>
<td>05L1R</td>
<td>1,271.83</td>
<td>3.74</td>
</tr>
<tr>
<td>05L1C</td>
<td>1,837.19</td>
<td>12.86</td>
</tr>
<tr>
<td>05L2R</td>
<td>1,113.46</td>
<td>6.39</td>
</tr>
<tr>
<td>05L2C</td>
<td>1,279.33</td>
<td>7.27</td>
</tr>
<tr>
<td>06L1R</td>
<td>1,514.38</td>
<td>65.09</td>
</tr>
<tr>
<td>06L1C</td>
<td>1,810.09</td>
<td>83.81</td>
</tr>
<tr>
<td>06L2R</td>
<td>1,115.20</td>
<td>112.84</td>
</tr>
<tr>
<td>06L2C</td>
<td>1,219.37</td>
<td>139.45</td>
</tr>
<tr>
<td>07L1R</td>
<td>1,410.35</td>
<td>148.03</td>
</tr>
<tr>
<td>07L1C</td>
<td>1,668.98</td>
<td>529.25</td>
</tr>
<tr>
<td>07L2R</td>
<td>1,276.15</td>
<td>505.56</td>
</tr>
<tr>
<td>07L2C</td>
<td>1,478.90</td>
<td>726.67</td>
</tr>
<tr>
<td>08L1R</td>
<td>1,926.49</td>
<td>3,600.01</td>
</tr>
<tr>
<td>08L1C</td>
<td>1,955.30</td>
<td>1,503.30</td>
</tr>
<tr>
<td>08L2R</td>
<td>1,499.73</td>
<td>3,081.50</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-------</td>
<td>----------</td>
<td>----------</td>
</tr>
<tr>
<td>08L2C</td>
<td>1,587.66</td>
<td>2,104.44</td>
</tr>
<tr>
<td>09L1R</td>
<td>1,895.64</td>
<td>3,600.01</td>
</tr>
<tr>
<td>09L1C</td>
<td>2,047.30</td>
<td>3,600.01</td>
</tr>
<tr>
<td>09L2R</td>
<td>1,716.89</td>
<td>3,600.01</td>
</tr>
<tr>
<td>09L2C</td>
<td>1,845.72</td>
<td>3,600.01</td>
</tr>
<tr>
<td>10L1R</td>
<td>2,758.10</td>
<td>3,600.02</td>
</tr>
<tr>
<td>10L1C</td>
<td>3,494.43</td>
<td>3,600.01</td>
</tr>
<tr>
<td>10L2R</td>
<td>1,989.60</td>
<td>3,600.03</td>
</tr>
<tr>
<td>10L2C</td>
<td>2,360.74</td>
<td>3,600.01</td>
</tr>
</tbody>
</table>

\(a\) Gap = (Objective bound – Current Incumbent Solution) / Current Incumbent Solution (%)

\(b\) Gap = (Objective form the solver – Best objective from GVNS) / Best objective from GVNS (%)
**REFERENCE**


