MIXED PICKUP AND DELIVERY VEHICLE ROUTING PROBLEM WITH TIME WINDOWS, SHIFTS AND MEAL BREAKS

by

Çiğdem Karademir

B.S., Industrial Engineering, Boğaziçi University, 2017

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Industrial Engineering

Boğaziçi University

2020

ACKNOWLEDGEMENTS

First of all, I would like to express my special thanks of gratitude to my supervisor Prof. Dr. Ümit Bilge for encouraging, motivating and giving confidence throughout this thesis. Her guidance and immense knowledge helped me in all the time of research and writing of this thesis. This work could not have been accomplished without her useful comments, remarks and engagement through the learning process of this master thesis.

Besides my advisor, I am grateful for having the opportunity to work with to Prof. Dr. Necati Aras for his insightful comments and guidance during the study and also being a jury member for my thesis committee. The door to Prof. Aras office was always open whenever I ran into a trouble spot or had a question about my research or writing. I would like to thank Assist. Prof. Hande Küçükaydın for being part of my thesis committee and for their valuable comments.

To Hande, Selin and Derya - thank you for always being enthusiastic, loving and supporting friends as well as sisters. We grew together and shaped our lives during the debates, dinners, and game nights as well as the parties and vacations.

I am eternally grateful to my dear family for their love, unlimited encouragement and trust in the path I have taken up to now. My life would have been much harder without their unconditional support. You brought me today and made me the person who I am. They deserve to take credit for all my accomplishments.

My appreciation also extends to my dear friends and fellow colleagues. Thanks to Feyyaz, Gizem, Orkun, Selin, Tarkan, Buğra and Elif for the great times we spent together, on-campuss and off-campus. You can make me laugh and happy during the coffee and lunch breaks. I also thank to my lab partners Gökalp, Mert, Bahadır, Gökçe and Ekin for all your help and friendly environment you provided in BUFAIM.

To Feyyaz, my dearest friend - thank you for being beside me all the undergraduate and graduate years. You are the definition of friendship in happiness and sadness. You give the point of view when I need it, and with your knowledge, you expand my vision into life and history in every subject.

Last but certainly not least, my deepest gratitude to Ahmet Pala. You are the joy of my life. Thank you for making me comfortable and valuable during all the years we have spent together. You have always been with me with your endless support in all the difficulties I faced during my graduate education.

I also thank TÜBİTAK for financially supporting me during my graduate studies under BIDEB-2210 program. Also, I would like to thank EKOL Logistic for providing this challenging problem to study in this thesis.

ABSTRACT

MIXED PICKUP AND DELIVERY VEHICLE ROUTING PROBLEM WITH TIME WINDOWS, SHIFTS AND MEAL BREAKS

Last mile delivery remains to be a focal point in the operations of logistics service providers. As more customers are willing to pay a premium price for receiving or sending their parcel within a specific time window, the Vehicle Routing Problem with Time Windows (VRPTW) has been an active research area. However, realistic problem features such as the existence of multiple working shifts and the meal breaks of the delivery personnel assigned to each shift have not received sufficient attention in the literature. The existence of these additional requirements gives rise to the VRPTW including Mixed Pickup, Delivery, Meal Breaks and Shifts (VRMPDTW-BS) which is the main focus of the present study. The fact that the time windows of some customers overlap with several shifts brings additional complexity since the problem cannot be decomposed with respect to shifts. We develop three methods for the solution of the VRPMPDTW-BS. The first method involves the formulation of a mixed-integer linear programming model, which can be solved to optimality for relatively small instances up to 25 customers. The main use of this model is to assess the quality of the solutions obtained by the proposed three-phase heuristic that is based on generating an initial solution and improving it using a large neighborhood search mechanism that monitors the positions of the nodes in the solution to learn their best place. With the help of node mobility concept, the initial solutions are improved in terms of the number of vehicles as well as the total distance by combining the proposed large neighborhood search with simulating annealing procedure. For larger instances with more than 1000 customers, we also devise another heuristic based on partitioning the customers into a number of clusters and then applying our heuristic for each cluster. The results obtained on instances of different sizes show that the proposed heuristic yields solutions of good quality within acceptable computation times for VRPMPDTW-BS. Additionally, the proposed heuristic is comparable to the state of art heuristics in the literature for different variants of Vehicle Routing Problem with Backhauls.

ÖZET

KARIŞIK DAĞITIM VE TOPLAMALI, ZAMAN KISITLI, VARDİYALI VE MOLALI ARAÇ ROTALAMA PROBLEMİ

Son alıcıya teslimat, hizmeti sağlayan lojistik firmalarının odak noktası olmaya devam etmektedir. Paketlerini belirli zaman aralığında teslim almak veya göndermek isteyen müşteri sayısı arttıkça Zaman Kısıtlı Araç Rotalama Problemi (VRPTW) etkin bir araştırma alanı olmaya devam etmektedir. Fakat birden fazla vardiyanın ve teslimat görevlilerinin mola saatlerinin var olması gibi gerçeğe yakın problem özellikleri literatürde gereken dikkati çekmemiştir. Bunlar gibi fazladan gereksinimler, bu tezin odak noktası olan Karışık Dağıtım ve Toplamalı, Zaman Kısıtlı, Vardiyalı ve Molalı Araç Rotalama Problemi'ni (VRPMPDTW-BS) açığa çıkarmıştır. Bazı müşterilerin zaman aralıklarının birden fazla vardiya ile örtüşmesi fazladan zorluk getirir. Çünkü problem vardiyalara göre ayrıştırılamaz. VRPMPDTW-BS çözümü için üç metot geliştirildi. İlk metot, 25 müşteriye kadar görece olarak daha küçük problemleri en iyi olarak çözen bir karışık tam sayılı lineer programlama modelinin formülasyonunu içerir. Bu modelin esas kullanımı, bir ilk çözüm üreten ve bu çözümü müşterilerin pozisyonlarını bir çözümdeki en iyi yerlerini öğrenmek için takip eden geniş komşuluk araması mekanizmasıyla iyileştiren üç aşamalı sezgisel yöntemin oluşturduğu çözümleri değerlendirmektir. Müşterilerin takibi kavramının yardımıyla, ilk çözümler araç sayısı bakımından olduğu gibi önerilen geniş komşuluk aramasının benzetimli tavlama ile birleştirilmesiyle toplam uzaklık bakımından da iyileştirilir. 1000 müşteriden fazla büyük problemler için, müşterileri belirli sayıda kümelere ayırmaya ve sonra her bir kümeye sezgisel yöntemimizin uygulanmasına dayanan başka bir sezgisel yöntem tasarlanmıştır. Farklı büyüklüklerdeki problemlerden elde edilen sonuçlar, önerilen sezgisel yöntemin VRPMPDTW-BS için kabul edilebilir hesaplama zamanı içinde iyi kalitede sonuçlar verdiğini göstermiştir. Ayrıca, önerilen sezgisel yöntem farklı Toplamalı Araç Rotalama Problemi çeşitleri için literatürde bulunan en gelişmiş sezgisel yöntemlere denk sonuçlar verebilmektedir.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

c	Cooling rate of the temperature in simulated annealing
d_i	Distance of node i to the *Pool*
Dev	Percent deviation
$MaxIter$	The maximum number of iteration in LNS-SNM
R	The number of nodes to remove in each *Destroy* procedure
T	Temperature in simulated annealing
T_i	*Tendency* value of node i
w	Upper limit of the percentage deviation of the distance in simulated annealing
α	*Tendency* update parameter of the nodes that cannot be inserted
β	*Tendency* update parameter of the nodes inserted into the same route and the same position
θ	*Tendency* update parameter of the nodes inserted into the same route but in a different position
γ	*Tendency* update parameter of the nodes inserted into a different node

LIST OF ACRONYMS/ABBREVIATIONS

CRBI	Cost Ratio Based Insertion
HVRPMPDTW	Heterogeneous Vehicle Routing Problem with Mixed Pickup, Delivery and Time Windows
LNS-SNM	Large Neighborhood Search with Smart Node Mobility
LNS-SNM-D	Large Neighborhood Search with Smart Node Mobility for Distance Minimization
LSP	Logistic Service Provider
MBS1	Mixed Integer Linear Programming Stage 1
MBS2	Mixed Integer Linear Programming Stage 2
MILP	Mixed Integer Linear Programming
MVRPBTW	Vehicle Routing Problem with Mixed Backhauls and Time Windows
VRP	Vehicle Routing Problem
VRPB	Vehicle Routing Problem with Backhauls
VRPBTW	Vehicle Routing Problem with Backhauls and Time Windows
VRPMPDTW	Vehicle Routing Problem with Mixed Pickup, Delivery and Time Windows
VRPMBTW	Vehicle Routing Problem with Mixed Backhauls and Time Windows
VRPMPDTW-BS	Mixed Pickup and Delivery Vehicle Routing Problem with Time Windows, Shifts and Meal Breaks
VRPTW	Vehicle Routing Problem with Time Windows
VRPTWLB	Vehicle Routing Problem with Time Windows and Lunch Breaks

1. INTRODUCTION

Last mile delivery has a central importance in the operations of logistic service providers (LSPs) worldwide. This importance will continue to increase in the future due to the boost in the number of people who make shopping online. It is estimated that e-commerce sales worldwide reach 3.54 trillion US dollars in 2020 and the same figure is expected to rise to 4.93 trillion US dollars in 2021 (Statista, 2020). 2.05 billion people buy at least one item online in 2020. As pointed out in the Digital Turkey 2019 E-commerce report, the percentage of people who make purchases on the internet was 68% in 2019, and the amount of money spent became 11.3 billion US dollars (hostingfacts.com, 2020). This corresponds to 39% increase from previous year in terms of local currency (Turkey, 2020). These estimates are made after the COVID19 epidemic and it is observed that they have been adjusted to even higher numbers when compared to the estimated values from the previous years for 2020. As a result, it appears that a growing trend will be observed in the number of home deliveries in the following years all over the world.

This study is motivated by the operational problems of an LSP operating in Turkey. This company offers many logistics services, and the line of business giving rise to the present study is the last mile delivery and pickup operations that aim to ensure timely service of products ordered by the customers at major e-tailers. Customers can request delivery (linehaul) or pickup (backhaul) service. All deliveries are loaded on the vehicles at the depot and all pickups are returned to the depot to be delivered the next day. There is no dependence between pickup and delivery customers or no precedence relationship between customers. The independence of customers enables us to serve pickups and deliveries in any order. Mixing pickups and deliveries reduces the logistics cost but may increase the material handling effort in the vehicle. Serving first deliveries then pickups is preferred in many logistic operations to reduce the effort in rearranging packages in the vehicle but places a significant constraint on routing problem and increases the operational costs. Regarding this trade-off, the LSP wants to operate as mixed pickups and deliveries. An emerging trend in this sector is the customers' behavior in receiving or sending the packages: more customers

are willing to pay a premium price for receiving or sending their parcel within a specific time window because they do not want the personnel to arrive when they are not at home. Besides this tendency in customer behavior, there are also some operational requirements to be considered by the LSP, which further complicate planning the activities of the LSP. First, the personnel of the company work in two shifts during the day. The first shift starts at 8:15 am and finishes at 3:45 pm, while the working hours of the second shift span the time between 2:45 pm and 10:15 pm. This means that the two shifts have overlapping service hours. The second requirement on the operational side is the existence of meal breaks in the shifts. The duration of the breaks is 30 minutes, and they take place between 12:00 pm and 12:30 pm in the first shift, and between 6:30 pm and 7:00 pm in the second shift. It is worth noticing that specific locations are not designated for meal breaks, and they can take place between two customer locations.

The goal of the LSP is to carry out the delivery and pickup operations with the smallest fleet of vehicles. This means that the company is less concerned with the total distance traveled by the vehicles. Most probably, this strategy is due to the fact that the fixed cost of a vehicle is much more significant than the reduction in the traveling distance or fuel cost that can be gained by increasing the fleet size by an additional vehicle. Hence, the problem addressed by the LSP is a variant of the Vehicle Routing Problem with Mixed Pickups and Deliveries and Time Windows (VRPMPDTW) which includes meal breaks and shifts. We refer to this problem as VRPMPDTW-BS.

As a matter of fact, the LSP defines several time windows for its customers and some of these time windows indeed overlap with both shifts. There are also time windows that entirely remain within the interval of one shift. The largest time window covers the daily working hours of the LSP. Figure 1.1 illustrates the nature of the time windows as well as the shifts. In this figure, Type 1 and Type 2 refer to the time windows remaining in a shift's interval whereas Type 0 time windows overlap with both shifts. It is worth mentioning that the existence of Type 0 time windows brings additional complexity to the solution of VRPMPDTW-BS since the problem cannot be decomposed with respect to the shifts. Therefore, in addition to the classical routing and service-start decisions that need to be made in VRPTW, decisions

related to customer-to-shift assignment decisions should also be taken into account. There is no extra cost on the waiting times. Notice that the waiting of a vehicle occurs when it arrives at a customer before the earliest start time of the service, that is the beginning of the time window.

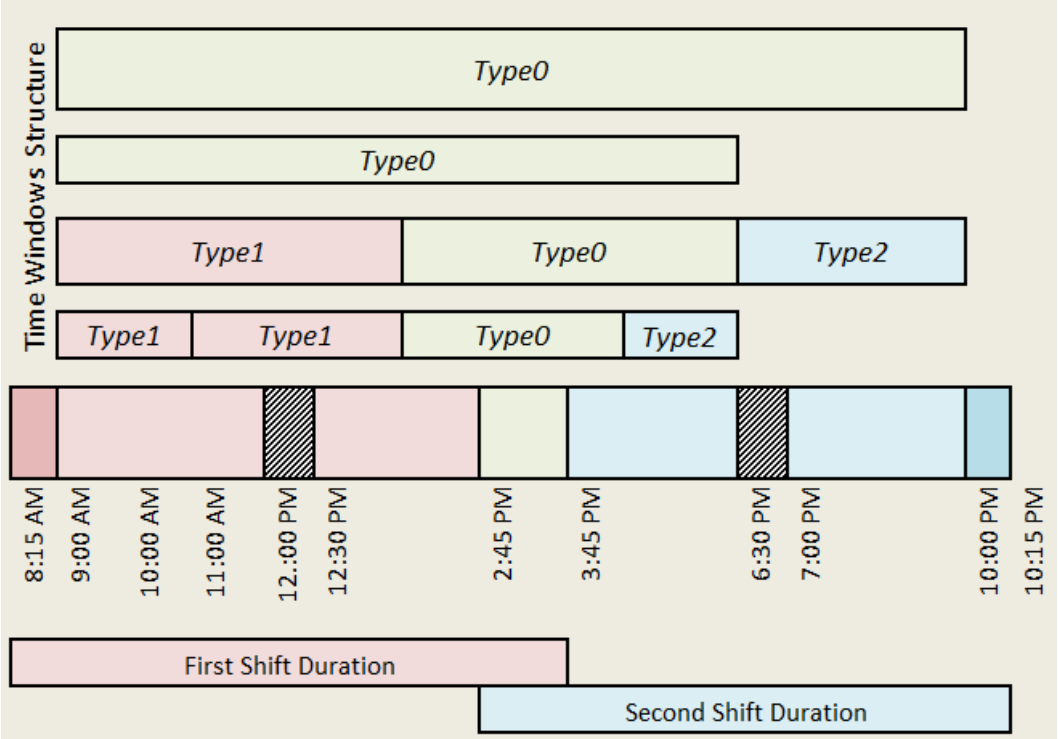

Figure 1.1. The structure of time windows and shifts.

Other characteristics of the problem dealt with in this study can be summarized as follows. There is a single depot, and the number of customers that need to be visited daily is above 1000. Moreover, the distances between the customers as well as between the depot and customers are asymmetric. Another important point worth mentioning based on the information provided by the LSP is that it is possible to assume that the fleet consists of homogeneous vehicles and that the vehicles are uncapacitated. The main reason of the latter assumption is the existence of time windows for customers, the duration of the shifts, meal breaks, and the non-zero service times at customers restrict the maximum number of customers that can be served by a vehicle in a shift. Moreover, the vehicles owned by the LSP have sufficient capacity to easily carry and serve all the items for customers whose number is limited due to aforementioned reasons.

We develop a heuristic method for the solution of the VRPMPDTW-BS, which turns out to be a rich VRP. This heuristic consists of three stages. In the first stage, a construction heuristic is utilized that is based on an extension of the classical Clarke-and-Wright savings heuristic (Clarke and Wright, 1964). This extension can handle all the restrictions of the problem and generate solutions of good quality within short computation times. The second stage involves a large neighborhood search that makes use of destroy and repair heuristics in an iterative way to reduce the number of vehicles as well as the total distance in the third stage. The proposed heuristic is called Large Neighborhood Search with Smart Node Mobility (LNS-SNM) and it utilizes the removal and insertion history of the nodes to learn their best positions for a given problem. During search iterations, LNS-SNM monitors the positions of nodes and updates their *tendency* values that measure the potential of nodes to relocate. LNS-SNM decides the set of nodes to remove by using *tendency* information to efficiently reduce the number of vehicles first, and then to reduce the total distance of vehicles. It is a powerful monitoring strategy for route reduction and also an efficient distance reduction tool when hybridized with a simulated annealing procedure.

In order to assess the performance of our heuristic, LNS-SNM, we develop a mixed-integer linear programming (MILP) model. To the best of our knowledge, it is the first mathematical programming model for VRPMPDTW-BS including meal breaks and shifts together. This model can be solved to optimality for problems up to 25 nodes. However, the size of the problem instances to be solved by the LSP on a daily basis exceeds 1000 customers, and an MILP model cannot be solved to optimality for problem instances of this size. Therefore, to assess the performance of the LNS-SNM for larger instances we opt for another approach that has the capability of solving larger-sized instances. The idea of this approach is based on a well-known technique, decomposition, which is frequently implemented in the operations research literature for large problem instances. We decompose a given VRPMPDTW-BS instance into smaller instances by means of k-means clustering, and apply our heuristic for the customer in each cluster.

The main contributions of this study can be summarized as follows:

- VRPMPDTW-BS, which includes multiple working shifts and meal breaks simultaneously, is defined for the first time in the literature.
- A mathematical programming formulation is developed for the VRPMPDTW-BS and optimally solved for instances up to 25 customers.
- A large neighborhood search, LNS-SNM, is proposed which can solve instances with more than 1000 customers.

The remainder of the thesis is organized as follows. Chapter 2 contains the literature review in the domain of Vehicle Routing Problem with Backhauls (VRPB). Section 3 defines the problem with its contraints and parameters and presents the mathematical models involved. Section 4 includes the proposed three-phase heuristic, LNS-SNM, while Chapter 5 consists of the computational results for the variants of VRPB and VRPMPDTW-BS. Chapter 6 finishes the thesis with concluding remarks and future research directions.

2. LITERATURE REVIEW

As in described in Introduction, this thesis focuses on VRPMPDTW-BS which is a rich variant of VRP with time windows that has not been studied in the literature before.

Pickup and delivery problems can be divided into three categories: *delivery-first, pickup-second*, *mixed pickup and deliveries* and *simultaneous pickups and deliveries* (Nagy and Salhi, 2005). In the last class, the requests of customers consist of picking up the package from a pickup location and delivering the package to a delivery location using the same vehicle. It imposes precedence constraints between pickups and delivery which is not the case for the problem studied in this thesis. Therefore, the first two classes will be considered in this study.

Vehicle routing problem with backhauls (VRPB) is a classical capacitated vehicle routing problem that incorporates backhaul customers by imposing several restrictions. Vehicles can serve first linehaul customers then backhaul customers if any backhaul customer is assigned. Routes having only linehaul customers is allowed but routes composing of only backhaul customers are forbidden. The objective of VRPB is to minimize total distance for a given number of vehicles, K. It needs to construct K many routes neither less nor more. Many heuristics are proposed to solve VRPB including population based heuristics, local search heuristics and neural netwok heuristics which can be found in a comprehensive survey conducted by Koç and Laporte (2018). All of them solve VRPB within 0.5% deviation or less on average.

Vehicle routing problem with backhauls and time windows (VRPBTW) is to construct the smallest fleet with the minimum total distance by serving the backhaul customers after the linehaul customers. It is similar to VRPB but in addition to hard time windows constraints for the nodes, VRPBTW permits routes having only backhauls in a route and the first objective is to minimize the number of vehicles. Gélinas et al. (1995) generate test instances from Solomon (1987) VRPTW benchmark test data for VRPBTW and they develop an exact

method using set partitioning model. Thangiah et al. (1996) use local search heuristic with λ-interchanges and 2-Opt edge exchanges. Potvin et al. (1996) present a greedy route construction and genetic algorithm for VRPBTW. Reimann and Ulrich (2006) solve VRPBTW using an ant colony optimization and discussed different backhauling strategies. Ropke and Pisinger (2006) develop a unified adaptive large neighborhood search for most of backhauling problems including VRPBTW by means of multiple destroy and repair heuristics. Tarantilis et al. (2013) design an adaptive path linking method by exploring multiple solutions and their trajectories using tabu search method with a novel diversity mechanism. Their method outperforms the existing methods used as benchmark in terms of cumulative number of vehicles and total distance.

Vehicle routing problem with mixed pickups, deliveries and time windows (VRPMPDTW) relaxes the first linehaul, then backhaul constraint on VRPBTW and it allows vehicles to serve deliveries and pickups in any order. Mixed vehicle routing problem with backhauls and time windows (MVRPBTW) is also used to refer this problem in the literature. The objective is still to minimize the number of vehicles first and then to minimize the total distance. There are few studies on VRPMPDTW. It is first studied by Kontoravdis and Bard (1995) using a greedy randomized adaptive search procedure (GRASP). Zhong and Cole (2005) develop a guided local search and solve VRPMPDTW with or without precedence constraints. Røpke and Pisinger (2004) propose a unified adaptive large neighborhood search for eight different types of vehicle routing problems with backhauls. They use six different destroy and two different repair heuristics and the quality of solutions to VRPMPDTW test problems are better than Kontoravdis and Bard (1995) and Zhong and Cole (2005) for all test problems. Tarantilis et al. (2013) also solve the same problem using an adaptive path linking method explained above. They report new best known solutions to existing VRPMPDTW test instances and outperform the existing heuristics.

Heterogeneous VRPMPDTW (HVRPMPDTW) is mostly solved by population based metaheuristics. Berghida and Boukra (2015) solve HVRPMPDTW using an Enhanced Biogeography-Based Optimization (EBBO) algorithm inspired from migration/emigration and mutation processes in the nature. The solutions generated using EBBO are better than

the published solutions of Belmecheri et al. (2013) that solve the same problem using particle swarm and ant colony optimization on the benchmark set generated from Solomon instances (Solomon, 1987). Wu et al. (2016) also study a population-based metaheuristic applied to HVRPMPDTW as in Belmecheri et al. (2013) and Berghida and Boukra (2015) but use a bi-objective optimization approached that aims at minimizing the total number of vehicles first, then the total distance of routes. Their labelling procedure is adopted from Cheung and Hang (2003) multi-attribute label matching algorithm and it is followed by a two-phase approach that minimizes the number of vehicles in the first stage, then the total distance of routes in the second stage using a modified version of ant colony optimization in each stage. Cheung and Hang (2003) develop simultaneous and sequential assignment algorithm apply iteratively in order to reduce the total distance of routes. Wu et al. (2016) achieve better solutions than Cheung and Hang (2003) on the same benchmark instances in terms of the total number of vehicles and the total distance of routes.

As mentioned in Coelho et al. (2016), there are relatively few studies in the literature that address VRPTW with lunch breaks (VRPTWLB) and no study incorporates lunch breaks within VRPMPDTW. A Waste Collection Vehicle Routing Problem with Time Windows is considered in Buhrkal et al. (2012), where the goal is to find routes for waste collection trucks so as to minimize the total distance of the routes. There exist two differences between this problem and the classical VRPTW. First, the trucks must also visit disposal sites within the routes whenever they are full to unload the waste they collected. It is possible that a vehicle visits the same disposal sites more than once. Second, drivers give two types of breaks: rest breaks and lunch breaks. After 4.5 hours of driving a rest break of 45 minutes must be given and a lunch break of minimum 30 minutes must be held once during the day. The authors propose an adaptive large neighborhood search heuristic to solve the problem. Although they also present an MILP, they do not solve it. Sahoo et al. (2005) focus on a similar waste collection problem that includes a single node for lunch break. An MILP is proposed for a simplified version of their problem, but no result is presented with respect to the solution of this mathematical model. An iterative two-phase algorithm is developed where an initial solution is generated in the first step which is then improved in the second step by an extended insertion algorithm as well as a simulated-annealing metaheuristic that makes use of the cross

exchange local search method proposed by Taillard et al. (1997). Another study which uses a single node representation for meal break is due to Sze (2011). The authors solve a real-life in-flight food loading operation as a multi trip VRPTWLB in which food items are loaded to aircrafts within their transit time. The loading personnel should have their meal breaks in lounge areas designated as a node that has a time window as large as possible to give a break. A mathematical model is formulated for the problem and a two-stage heuristic is proposed. The heuristic solutions are better than those of the company, but relatively worse than the optimal solutions.

The same problem is investigated by Kim et al. (2006) and solved by means of a clustering-based route building algorithm that is implemented with and without simulated annealing. Benjamin and Beasley (2010) also study the same waste collection problem as Kim et al. (2006). They present three different metaheuristics based on variable neighborhood search, tabu search, and variable neighborhood search with tabu search. These metaheuristics provide solutions of similar quality, but the average distance of these solutions are smaller and the number of vehicles is higher than those of Kim et al. (2006). Therefore, to reduce the number of vehicles they introduce a vehicle reduction procedure which gives rise to the same number of vehicles. The VRPTWLB is addressed in Coelho et al. (2016), where both a new mathematical formulation and multi-start randomized local search heuristic are introduced for it. The developed MILP can only provide optimal solutions for very small instances with five and 10 customers. An instance with 20 customers can be solved with 52% optimality gap in one-hour computation time. The reason lies in the fact that the MILP formulation does not use binary variables to indicate the customer nodes after which a lunch break is given, rather the number of customer nodes is doubled in order to represent the break nodes, which significantly increases the number of routing variables. An important result of the thesis is the demonstration of integrating drivers' breaks into the model. The authors show that solving the VRPTWLB without explicitly taking into account the breaks and insert them into the obtained routes in a post-processing step may generate infeasible solutions. A unified hybrid genetic search metaheuristic is devised by Vidal et al. (2014) to solve the VRPTWLB and other rich VRP problems. Lunch breaks can be given in a route at a location that can be selected from among a set of potential sites or the location of the break can be unrestricted.

The building blocks of this heuristic that are problem-independent are unified local search, genetic operators, and diversity management methods, while there also exist a limited number of problem-specific components.

Within the context of waste collection including multiple depots and intermediate facilities, a mathematical model is formulated by Markov et al. (2016) in which breaks are defined on arcs and the number of vehicles is fixed. Exact solutions are obtained by solving this model up to 15 customers while multiple neighborhood search instances with up to 288 customers and six intermediate facilities.

3. MATHEMATICAL MODEL

As pointed out earlier, the company has two objectives that are hierarchical in nature. The objective of minimizing the number of vehicles has priority over the other objective which is related to the minimization of the total traveling distance. Since the problem is a rich VRP with many constraints including time windows, meal breaks, and working shifts, it is not clear how many vehicles would be required to obtain a feasible solution for an instance of VRPMPDTW-BS. Note that, distance minimization does not guarantee the minimum number of vehicles, for a given problem instance in general. For example, the total distance with a larger number of routes might be smaller than the total distance with a smaller number of routes. Therefore, we first formulate an MILP model in order to determine the minimum number of vehicles required, which is referred to as MBS1. After having obtained the minimum number of vehicles from the optimal solution of MBS1, we utilize another MILP model called MBS2 which has the objective of minimizing the total distance traveled by the vehicles. In other words, the second model takes as an input parameter the number of vehicles necessary for the operation. In formulating the two models, we use the following index sets, parameters, and decision variables.

We let $\mathcal{N}$ denote the set of delivery and pickup customer nodes at which a service has to be performed. All vehicles depart from a depot indexed as node 0, and the set $\mathcal{N}_0 = \mathcal{N} \cup \{0\}$ represents all the customers as well as the depot. The sets $\mathcal{D}$ and $\mathcal{P}$ represent the delivery nodes and pickup nodes respectively.

Associated with all customer nodes, there exist earliest and latest service starting times which are denoted by parameters e_i and ℓ_i, respectively. It is worth noticing that these parameters designate the time interval during which service can begin at a customer node, but not the arrival time of the vehicle at the customer node. Whenever a vehicle arrives at a customer before e_i, then the vehicle has to wait until e_i for the service to take place. There may exist waiting times which do not incur any penalty or cost for the company. The service time at a customer node is s_i minutes for every $i \in \mathcal{N}$. Among other parameters, we can

mention parameter t_{ij} that represents the traveling time from node $i \in \mathcal{N}_0$ to node $j \in \mathcal{N}_0$. q_i is the delivery or pickup demand for customer node $i \in \mathcal{N}$. c_{ij} denotes the cost of traveling from node $i \in \mathcal{N}_0$ to node $j \in \mathcal{N}_0$.

The delivery personnel of the company work in two shifts which are indexed as $\mathcal{K} = \{1, 2\}$. The first shift starts at 8:15 am and finishes at 3:45 pm, while the working hours of the second shift last from 2:45 pm to 10:15 pm. The set $\mathcal{N}_k$ represents the delivery and pickup nodes that can be served in shift k. If the horizon of the time window of a node intersects or overlaps with the operational horizon of shift k, the node is included in $\mathcal{N}_k$. Type$k$ customers who can be served only in shift k and Type0 customers who can be served in both shifts belong to $\mathcal{N}_k$. The starting and finishing times of the shifts are represented by the parameter h_k^s and h_k^f, respectively for $k = 1, 2$. The time unit is set to minutes. This implies that $h_1^s = 0$ and $h_1^f = 450$ for the first shift and $h_2^s = 390$ and $h_1^f = 840$ for the second shift. Each shift has a meal break of 30 minutes. The meal breaks must take place between 12:00 pm and 12:30 pm for the first shift and between 6:30 pm and 7:00 pm for the second shift. We use the parameters b_k for the duration of the shift k. As a matter of fact, it is possible to use a single parameter for both shifts as they are equal. Parameters $b_1^s = 225$, $b_1^f = 255$ and $b_2^s = 615$, $b_2^f = 645$ denote respectively the starting and finishing times of the meal breaks, for the shifts. There should not be any service during break periods. $\mathcal{N}_b$ is the set of delivery and pickup nodes that can be served before the meal break in any shift. In other words, these nodes are candidate nodes that the driver can take the meal break. If the first node visited by any vehicle has its earliest service start time, e_i, after the break start time of the vehicle's shift, the driver takes break before the service start. The driver can start service at the node immediately after the end time of the break and there is no need to assign a meal break for that vehicle anymore. Operationally, a driver could start its service after the meal break or could return to the depot before the meal break for the minimizing total distance. In any case, the driver can have the meal break at the depot and no service is given during the meal break.

3.1. Modelling VRPMPDTW-BS

This section presents a mixed-integer linear model (MILP) for VRPMPDTW-BS defined in the previous section. It is a rich variant of classical Vehicle Routing Problem with Time Windows (VRPTW) including many operational constraints. The model aims to find the minimum number of vehicles with the lowest total distance in the presence of following limitations:

i. Each node is visited once.
ii. Each node is visited within its time window.
iii. Pickup and delivery nodes can be visited in any order.
iv. Each vehicle performs exactly one route and serves in only one of the shifts.
v. The capacity of any vehicle is not violated at any time.
vi. Delivered items are loaded at the depot and collected items are returned to the depot.
vii. Each vehicle takes exactly one meal break within the meal break time window specified by its shift.
viii. Each vehicle departs from the depot and returns to the depot within its shift duration.

Sets and parameters used in the model can be summarized as follows:

Index sets and parameters:

$\mathcal{D}$: The set of delivery nodes

$\mathcal{P}$: The set of pickup nodes

$\mathcal{N}$: The set of pickup and delivery nodes, $N = D \cup P$

$\mathcal{N}_0$: The set of delivery nodes, pickup nodes and the depot, $N_0 = N \cup \{0\}$

$\mathcal{K}$: The set of shifts

$\mathcal{N}_b$: The set of delivery and pickup nodes that can be served before the meal break for any shift $k \in \mathcal{K}$

Parameters:

c_{ij}	:	Cost of traveling from node $i \in N_0$ to node $j \in N_0$
t_{ij}	:	Traveling time from node $i \in N_0$ to node $j \in N_0$
e_i	:	Earliest time service can start at node $i \in N$
ℓ_i	:	Latest time service can start at node $i \in N$
s_i	:	Service time for customer $i \in N$
q_i	:	Demand for customer $i \in N$
Q	:	Capacity of a vehicle
b_k	:	Duration of the meal break for shift $k \in \mathcal{K}$
b_k^s	:	Starting time of the meal break for shift $k \in \mathcal{K}$
b_k^f	:	Finishing time of the meal break for shift $k \in \mathcal{K}$
h_k^s	:	Starting time of shift $k \in \mathcal{K}$
h_k^f	:	Finishing time of shift $k \in \mathcal{K}$
M_{1ijk}	:	ℓ_i - e_j + s_i + t_{ij}+ b_k for $k \in \mathcal{K}$, $i \in N_k$, $j \in N_k$
M_{2ik}	:	ℓ_i + s_i for $k \in \mathcal{K}$, $i \in N_k$
M_{3k}	:	b_k^f for $k \in \mathcal{K}$
M_{4ik}	:	ℓ_i+ s_i + t_{i0} - h_k^f for $k \in \mathcal{K}$, $i \in N_k$
M_{5ik}	:	h_k^s - e_i + t_{i0} for $k \in \mathcal{K}$, $i \in N_k$

The final five parameters that start with M introduce big numbers to the model and defined as tight as possible without disturbing the feasibility of the problem.

The decision variables are defined as follows. Binary variable $X_{ijk} = 1$ if a vehicle travels from node i to node j in shift k, it is zero otherwise. Note that we are not using an index for the vehicle since the vehicles are homogeneous. To build an easily-to-read model, X_{ijk} is defined as zero if visiting the node i immediately after the node j in shift k is not feasible with respect to time windows constraints. It represents infeasible arcs. If the node i can be served only in the first shift, all the flow decision variables in other shift(s) are set to zero. Another case is that the earliest service time of the node j is earlier than the latest service time of the node i and it is infeasible to visit the node j immediately after the node i in any

shift. Another binary variable is used to determine the timing of meal breaks. Namely, $Y_i = 1$ if a meal break is given by a vehicle immediately after visiting node i, it is zero otherwise. There exist three continuous decision variables in the model. T_i denotes the starting time of the service at node i. D_i and P_i represent respectively the total load of deliveries and total load of pickups on the vehicle immediately after visiting node i. All the decision variables with their explanations used in the modelling VRPMPDTW-BS are provided below and they are followed by model MBS1.

Decision variables:

X_{ijk} : 1 if a vehicle travels from node $i \in \mathcal{N}_0$ to node $j \in \mathcal{N}_0$ in shift $k \in \mathcal{K}$; 0 otherwise

Y_i : 1 if a meal break for a vehicle occurs immediately after visiting node $i \in \mathcal{N}$; 0 otherwise

T_i : Starting time of the service at node $i \in \mathcal{N}$

P_i : Total pickup load on the vehicle after visiting node $i \in \mathcal{N}$

D_i : Total delivery load on the vehicle after visiting node $i \in \mathcal{N}$

Model MBS1:

$$\min V = \sum_{j \in \mathcal{N}} \sum_{k \in \mathcal{K}} X_{0jk} \tag{3.1}$$

subject to

$$\sum_{\substack{j\in\mathcal{N}_0\\ j\neq i}}\sum_{k\in\mathcal{K}} X_{ijk} = 1 \qquad i \in \mathcal{N} \tag{3.2}$$

$$\sum_{\substack{j\in\mathcal{N}_0\\ j\neq i}}\sum_{k\in\mathcal{K}} X_{jik} = 1 \qquad i \in \mathcal{N} \tag{3.3}$$

$$\sum_{\substack{j\in\mathcal{N}_0\\ j\neq i}} X_{ijk} = \sum_{\substack{j\in\mathcal{N}_0\\ j\neq i}} X_{jik} \qquad i \in \mathcal{N}, k \in \mathcal{K} \tag{3.4}$$

$$\sum_{k\in\mathcal{K}} X_{ijk} \leq 1 \qquad i \in \mathcal{N}_0, j \in \mathcal{N}_0 \tag{3.5}$$

$$\sum_{i\in\mathcal{N}} X_{0ik} = \sum_{i\in\mathcal{N}} X_{i0k} \qquad k \in \mathcal{K} \tag{3.6}$$

$$\sum_{i\in\mathcal{N}_b} Y_i = \sum_{k\in\mathcal{K}}\sum_{i\in\mathcal{N}_b} X_{0ik} \tag{3.7}$$

$$e_i \leq T_i \leq \ell_i \qquad i \in \mathcal{N} \tag{3.8}$$

$$T_i + s_i + t_{ij} + b_k Y_i \leq T_j + M_{1ijk}(1 - X_{ijk}) \qquad i \in \mathcal{N}, k \in \mathcal{K} \tag{3.9}$$

$$T_i + s_i \leq b_k^s + M_{2ik}(2 - Y_i - \sum_{j\in\mathcal{N}_0} X_{ijk}) \qquad i \in \mathcal{N}, j \in \mathcal{N}, k \in \mathcal{K} \tag{3.10}$$

$$T_j \geq b_k^f Y_i - M_{3k}(2 - Y_i - X_{ijk}) \qquad i \in \mathcal{N}, j \in \mathcal{N}, k \in \mathcal{K} \tag{3.11}$$

$$T_i + s_i + t_{i0} \leq h_k^f + M_{4ik}(1 - X_{i0k}) \qquad i \in \mathcal{N}, k \in \mathcal{K} \tag{3.12}$$

$$h_k^s + t_{0i} - M_{5ik}(1 - X_{0ik}) \leq T_i \qquad i \in \mathcal{N}, k \in \mathcal{K} \tag{3.13}$$

$$D_j \leq D_i - q_j + Q(1 - \sum_{k\in\mathcal{K}} X_{ijk}) \qquad i \in \mathcal{N}_0, j \in \mathcal{D} \tag{3.14}$$

$$D_j \leq D_i + Q(1 - \sum_{k\in\mathcal{K}} X_{ijk}) \qquad i \in \mathcal{N}_0, j \notin \mathcal{D} \tag{3.15}$$

$$D_i \leq Q(1 - \sum_{k\in\mathcal{K}} X_{i0k}) \qquad i \in \mathcal{N} \tag{3.16}$$

$$P_j \geq P_i + q_j - Q(1 - \sum_{k\in\mathcal{K}} X_{ijk}) \qquad i \in \mathcal{N}_0, j \in \mathcal{P} \tag{3.17}$$

$$P_j \geq P_i - Q(1 - \sum_{k\in\mathcal{K}} X_{ijk}) \qquad i \in \mathcal{N}_0, j \notin \mathcal{P} \tag{3.18}$$

$$P_0 = 0 \tag{3.19}$$

$$D_i + P_i \leq Q \qquad i \in \mathcal{N} \tag{3.20}$$

$$X_{ijk} \in \{0,1\} \qquad i \in \mathcal{N}_0, j \in \mathcal{N}_0, k \in \mathcal{K} \tag{3.21}$$

$$Y_i \in \{0,1\} \qquad i \in \mathcal{N}_b \tag{3.22}$$

$$Y_i = 0 \qquad i \notin \mathcal{N}_b \tag{3.23}$$

$$T_i \geq 0 \qquad i \in \mathcal{N} \tag{3.24}$$

$$0 \leq D_i \leq Q \qquad i \in \mathcal{N}_0 \tag{3.25}$$

$$0 \leq P_i \leq Q \qquad i \in \mathcal{N}_0 \tag{3.26}$$

In this formulation, Equation (3.1) represents the objective function, which minimizes the number of vehicles used or equivalently the number of resulting tours. Equation (3.2) imply that there is one departure from every customer node and it occurs in one of the shifts. By the same token, Equation (3.3) ensure that there is one arrival to every customer node and it occurs in one of the shifts. In other words, these two constraints guarantee that a customer is visited only once over two shifts. Equation (3.4) are balance equations and ensure that if a customer is visited in shift k, then there must exist an incoming and outgoing arc to that customer in shift k. Equation (3.5) make sure that an arc (i, j) can either be traversed by a vehicle in one shift or not at all. Equation (3.6) state that in a shift the number of vehicles leaving the depot must be equal to the number of vehicles entering the depot. Equation (3.7) state that the number of meal breaks must be as large as the number of vehicles used excluding the vehicles whose first service start time is greater than the meal break starting time of the vehicle's shift. Equation (3.8) ensure that the starting time of the service at a customer node is within the time window determined by the earliest and latest times. Equation (3.9) relate the service start times between two consecutive customer nodes. If customer j is visited after customer i by a vehicle, then the service start time at node j cannot be earlier than the finish time at node i given as $T_{ik} + s_i + t_{ij}$ delayed by the meal break, if any. If there is no travel from node i to node j in shift k, then $X_{ijk} = 0$ and Equation (3.9) becomes redundant. Equation (3.10) imply that if a meal break occurs after visiting node i ($Y_i = 1$), then the service at that node must be finished before the starting time of the break in shift k if node i is visited in shift k. To put it differently, customer node i has to be the last node served before the meal break starts. If there is no break immediately after node i or it is not served in shift k,

Equation (3.10) becomes redundant. Equation (3.11) set a condition for the service start time at customer node j that is visited right after the meal break. Namely, the starting time of the service at node j cannot be earlier than the finishing time of the meal break in shift k if this node is visited immediately following node i after which the break is given. Equations (3.10) and (3.11) together do not allow more than one meal break on any route. Equation (3.12) act as maximum tour duration constraints. If customer node i is the last node visited of a tour in a shift, then $X_{i0k} = 1$ and the arrival time at the depot should not exceed the finishing time of the shift. In a similar vein, Equation (3.13) require that the starting time of the service at the customer immediately visited after the depot be later than the starting time of the shift. Equation (3.14) reduce the total delivery load on the vehicle by q_j after visiting a delivery customer node j immediately after node i. If node j is not a delivery customer, Equation (3.15) impose that the total delivery load on the vehicle after visiting node j should be as large as the total delivery load on the vehicle after visiting the node i which is the predecessor of node j. These constraints enable us to assign a total delivery load on the vehicle after visiting node i which is the sum of all delivery demands that need to be served after node i in the same tour. Equation (3.16) state that if node i is the last customer on a tour, the total delivery load on the vehicle after visiting node i must be equal to 0 since the vehicle performs all deliveries on its tour. In the same manner, Equation (3.17) increase the total pickup load on the vehicle by q_j after visiting a pickup customer node j immediately after node i. If node j visited after node i in a shift is not a pickup customer, then Equation (3.18) impose that the total pickup load on the vehicle after visiting node j does not decrease since all collected items are returned to the depot. Setting the total pickup load on a vehicle to zero by Equation (3.19), the total pickup load on a vehicle increases by q_i only if a pickup node is visited. It is a valid assignment since items are collected from the customers not loaded at the depot. Equation (3.20) ensure that the sum of total pickup and delivery load on a vehicle should not exceed the capacity of the vehicle given as Q. Equations (3.21), (3.23) and (3.22) restrict X_{ijk} and Y_i to be binary decision variables, respectively. Finally, Equations (3.24), (3.25), and (3.26) define T_i, D_i and P_i as continuous variables.

The optimal solution of MBS1 provides the minimum number of tours (vehicles) $V^* = \sum_{j \in \mathcal{N}} \sum_{k \in \mathcal{K}} X^*_{0jk}$ to have a feasible solution for a given problem instance without paying attention to the total distance traveled by the vehicles. Now, we make use of MBS2 which takes the number of routes as input and minimizes the total distance traveled, which we designate with Z. Hence, the mathematical programming formulation of MBS2 can be written as

Model MBS2:

$$\min Z = \sum_{i \in \mathcal{N}_0} \sum_{j \in \mathcal{N}_0} \sum_{k \in \mathcal{K}} c_{ij} X_{ijk} \tag{3.27}$$

subject to

$$\sum_{i \in \mathcal{N}} \sum_{k \in \mathcal{K}} X_{0ik} \leq V^* \tag{3.28}$$

and Equations (3.2)–(3.26) (3.29)

We remark that Equation (3.28) is not redundant since the minimum total distance traveled may be obtained with a number of vehicles larger than V^* in the absence of this constraint. Recall that the company policy dictates that the objective of minimizing the number of vehicles takes precedence over the objective of minimizing the distance.

4. SOLUTION METHODOLOGY

A Large Neighborhood Search with Smart Node Mobility (LNS-SNM) is proposed to solve VRPMPDTW-BS. First, an initial solution is generated. Then LNS-SNM is used for reducing the number of vehicles in the solution. Finally, LNS-SNM is re-used after hybridizing with simulated annealing for improving the total distance of the solution having the minimum number of vehicles found in route reduction step.

4.1. Initial Route Generation

Most of the route construction heuristics for VRPTW in the literature use an insertion criterion derived mainly as an extension of the insertion heuristic developed in Solomon (1987) inspired by Clarke and Wright savings algorithm Clarke and Wright (1964). This insertion heuristic computes savings in total distance by regarding a limit for total waiting time of a route for feasibility while merging routes. According to the survey conducted by Bräysy and Gendreau (2005), the most successful construction heuristic for VRPTW is I1 sequential heuristic developed by Ioannou et al. (2001). The insertion criterion is divided into three metrics to measure the impact of insertion of i. Those metrics are the impact of insertion of i on itself (own impact), on the customers in the route that i is inserted (internal impact), and on the non-routed customers (external impact). Total impact is calculated as a weighted combination of these impacts; and uses six different parameters. It requires parameter tuning and they could not offer a clear explanation of the relationships between the parameters and problem types. While I1 produces good solutions for randomly distributed customers having tight time windows, it causes excessive run times for the real case application having over 1000 nodes studied in this thesis where we have a larger number of randomly clustered customers, and wide time windows. Another well-known construction heuristic is the basic greedy heuristic which inserts the cheapest customer. However, it constructs too many routes in our case and thus turns out to be inefficient for the route elimination phase of our heuristic. Therefore, a new insertion heuristic referred to as Cost Ratio Based Insertion Heuristic (CRBI) is devised to construct an initial solution for VRPMPDTW-BS.

A main decision to make besides the usual decisions for VRPMPDTW-BS is assignments of customers to the shifts. Two shifts with overlapping times exist and there are some wide time windows such that most customers can be served in both shifts. To differentiate customers by their shifts and avoid a type incompatibility, each customer and each vehicle is assigned to a type which indicates their shifts and all moves are forbidden between incompatible types. Three types can occur in this problem:

- Type 0: If the time window of a customer overlaps with both shifts or equivalently both shifts can serve it. This occurs if e_i starts before the first shift ends ($e_i \leq h_1^f$) and ℓ_i is after the second shift starts ($\ell_i \geq h_2^s$) for customer i.
- Type 1: If the time window of a customer overlaps with the first shift or equivalently only the first shift can serve it. This occurs if ℓ_i is before the second shift starts ($\ell_i \leq h_2^s$) for customer i.
- Type 2: If the time window of a customer overlaps with the second shift or equivalently only the second shift can serve it. This occurs if e_i is after the first shift ends ($e_i \geq h_1^f$) for customer i.

A vehicle can only serve in one shift. Its required departure time from the depot, arrival time at the depot and meal break time are all specified by its shift. A vehicle is associated with a type determined by the dominating customer type among its customers. Type 1 and Type 2 dominate Type 0. If a vehicle has at least one Type 1 customer among its Type 0 customers, its type is designated as Type 1. The same is true for Type 2. After each modification in a vehicle's route, its type is updated. A vehicle is Type 0 only if all its customers are of Type 0. A Type 0 vehicle is feasible in terms of time windows, tour duration, capacity and meal break if it can serve all the customers according to one of the shifts. It can be also said that the feasibility of a Type 0 vehicle is checked according to both shifts' specifications. If it is feasible in at least one of the shifts, it is accepted and its type is still 0. Its type is not updated as the defining shift type in which it is feasible just because all moves are performed according to type compatibility. A vehicle cannot have customers whose types are 1 and 2 simultaneously. Thus, Type 1 and Type 2 are incompatible in terms of shift. Any move between Type 1 customer and Type 2 customer, Type 1 vehicle and Type 2 vehicle, Type 1

vehicle and Type 2 customer, and Type 2 vehicle and Type 1 customer is forbidden due to type incompatibility.

While the insertion of a route is only possible at the end or beginning of another route in Clarke and Wright (1964), our construction heuristic CRBI inserts a route into any position in another route when feasible and compatible. We remark that the sequence of nodes in route u is preserved while inserting it into route v. To tackle the asymmetry in distances, reversals of routes are also considered. This means a route can be inserted into any position in another route after reversing its sequence.

For example when we want to insert $Route(u)$: 1-2 into $Route(v)$: 3-4, candidates of Route(u,v) after merging are obtained as follows: 1-2-3-4, 3-1-2-4, 3-4-1-2 - 2-1-3-4 - 3-2-1-4, 3-4-2-1, 1-2-4-3, 4-1-2-3, 4-3-1-2, 2-1-4-3, 4-2-1-3, and 4-3-2-1.

In addition to handling asymmetry, this insertion mechanism also helps to connect customers that are close spatially and temporally. For example, let us say that customers i and j are currently on different routes but spatially close to each other. Customer i whose time window is wide is currently at the beginning of its route while earliest start time of j's time window is late. In this case j cannot be inserted next to i because of time window infeasibility. However, reversal of the route that i belongs to can be feasible allowing i and j to be connected successfully. At the beginning of CRBI, many insertions are possible while after merging routes one by one, possible insertions decrease dramatically since reversals and most of positions in other routes are not feasible due to time windows.

For each candidate $Route(u, v)$ generated as above, CRBI calculates a measure called $r(u, v)$, which represents the cost of the route per node visited. For each route pair (u, v), the best insertion cost $r^*(u, v)$ is selected among all $r(u, v)$ values which represent the insertion costs. An $n \times n$ matrix called Ratio Matrix is created and maintained throughout the construction phase. The uth row and vth column of the Ratio Matrix contains $r^*(u, v)$. After all entries are calculated at the beginning of CRBI, some of them are updated as necessary during the subsequent iterations. Alternative cost definitions can be designed

which perform differently with respect to the problem characteristics such as the distribution of customers' locations, the length of scheduling period, tightness and distribution of time windows. Solomon (1987) created six different sets of benchmark instances regarding those factors. We defined the following three cost ratios and tested them on Solomon's benchmark instances as well as ours.

$$r_1(u,v) = \frac{Total\_distance}{Number\ of\ nodes\ in\ the\ candidate\ Route(u,v)} \tag{4.1}$$

$$r_2(u,v) = \frac{Total\_distance + Total\_waiting\_time}{Number\ of\ nodes\ in\ the\ candidate\ Route(u,v)} \tag{4.2}$$

$$r_3(u,v) = \frac{Tour\_duration - Service\_start\_time\_at\_first\_node}{Number\ of\ nodes\ in\ the\ candidate\ Route(u,v)} \tag{4.3}$$

$r_1(u,v)$ performs best when the instance contains clustered customers while $r_2(u,v)$ is efficient when customer locations are uniformly distributed with some additional clusters. In our instances where there exist several distant customers besides a mix of uniformly distributed and clustered locations, $r_3(u,v)$ becomes the winner. Therefore, we use $r_3(u,v)$ in CRBI.

After calculating $r_3(u,v)$ for each candidate $Route(u,v)$, the best candidate $Route^*(u,v)$ with the minimum ratio $r_3^*(u,v)$ is selected and recorded as the entry in the uth row and vth column of a matrix. Some entries of this matrix may not exist due to the feasibility constraints in terms of time windows, route duration, and shift compatibility. The next step is to select the best (u^*, v^*) pair with the smallest $r_3^*(u,v)$ to perform the insertion operation. CRBI heuristic is summarized in Figure 4.1.

1: Input: Distance matrix, customer information including ID and Time Window, Shift information including shift starting and ending times, and meal break times
2: *Initialization:* Make n routes: 0 -> i -> 0 for each customer i. The type of each route is equivalent to the customer type it serves
3: *Ratio Matrix Construction:* Compute $r_3^*(u, v)$ for each u and v routes pair if feasible
4: **while** There exists at least one feasible entry in Ratio Matrix **do**
   4.1 *Best Insertion Decision:* Find the minimum $r_3^*(u, v)$ and perform the corresponding insertion operation. Assign vehicle type to newly created route with respect to dominant shift.
   4.2 *Update Ratio Matrix:* Delete all entries related to route u^* and route v^* from Ratio matrix, which are merged in step 4.1. Add a new row and column for the merged $Route(u^*, v^*)$ into Ratio matrix by computing $r(k, Route(u^*, v^*))$ and $r(Route(u^*, v^*), k)$ for each route k.
5: **end while**
6: **return** Final routes

Figure 4.1. Cost Ratio Based Insertion Heuristic Algorithm

4.2. A Large Neighborhood Search with Smart Node Mobility

In this section, LNS-SNM is presented as a route and distance reduction search heuristic based on large neighborhood search. It consists of a single destroy and repair heuristic despite the usage of multiple number of heuristics in the literature. A new concept is used for choosing nodes for the destroy heuristic. *Tendency* for mobility can be defined as the willingness of a node to change its location or the potential of a node to be successfully relocated. Initialization of *tendency* values of the nodes for a given problem instance can be customized with respect to the problem type, specifically according to the limitations of nodes to relocate. These limitations might be the width of their time windows, the sparseness of the network or the shift type of the nodes in the existence of multiple working shifts. For different types of

problems, the initialization will be discussed in the computational results.

A solution can be improved by changing the route of a node or changing its position in the assigned route. By consecutive removals and insertions of a set of nodes, LNS-SNM learns the *tendency* values of the removed nodes by checking their insertion places regarding the route or the position in the route if the route is the same. Regarding their successes to be inserted in an iteration and their insertion positions, tendency values of nodes are multiplied by different parameters that varies between 0 and 1. Four different cases can occur:

i. If the route and the position in the route of the node do not change, its *tendency* value is multiplied by β.
ii. If the route of the node does not change but the position of the node in the route differs, its *tendency* value is multiplied by θ.
iii. If the route of the node changes, its *tendency* value is multiplied by γ.
iv. If the node cannot be inserted, its *tendency* value is multiplied by α.

The largest parameter is γ because multiple routes exist for the node. θ which stands for the position change in the same route is slightly less than γ to account for less ability to move. β is sharply less than the γ and θ since neither route nor the position of the node differs. The smallest parameter is α that is used for the node that stays in *Pool* over and over after insertions to prevent it from selection in destroy heuristics frequently. It enables us to keep the nodes, which are hard to place again, in the solution. By using this justification, it is decided that $\gamma > \theta > \beta > \alpha$.

The first objective of VRPMPDTW-BS is minimizing the number of vehicles. Route reduction is achieved by destroying a whole route and trying to find feasible insertions for the nodes in the destroyed route. A list for the unserved nodes, *Pool*, is kept throughout LNS-SNM. Destroy heuristic adds more nodes to *Pool* to make room for all unserved nodes including the removed nodes by the destroy heuristic. There is predefined number, R, of nodes to be removed in this fashion. A single repair heuristic which is specified as an input to the algorithm is used within LNS-SNM to insert the nodes in the *Pool* during the iterations.

Whenever the *Pool* is emptied, a full solution is obtained with a fewer number of vehicles. Then, the algorithm fills the *Pool* with the nodes of another route to try decreasing the number of routes by one. Route destroy and iterative removals and insertions are performed until the *Pool* of unserved nodes remain non-empty for a limited number of consecutive iterations.

The outline of LNS-SNM for route reduction is presented in Figure 4.2.

1: Input:α, β, θ, γ, MaxIter, R, $RepairHeuristic$, a feasible solution
2: Initialize tendency values, last routes and last positions for all nodes. Set *Iter* =0
3: Initialize *Pool* with the nodes in the smallest route having the least number of nodes in the current solution. Remove the route from the solution.
4: **while** *Iter*<*MaxIter* **do**
5: Increase *Iter*.
6: Perform *Destroy(Pool,R)*.
7: Perform *Repair Heuristic* for the nodes in *Pool*. Keep a list for the nodes inserted as *InsertedNodes*.
8: **for each** node in *InsertedNodes* **do**
9: **if** The node is inserted into the same route and the same position **then**
10: Multiply its tendency by β.
11: **end if**
12: **if** The node is inserted into the same route but in a different position **then**
13: Multiply its tendency by θ.
14: **end if**
15: **if** The node is inserted into a different route **then**
16: Multiply its tendency by γ.
17: **end if**
18: **end for**
19: **for each** Node in *Pool* **do**
20: Multiply its tendency by α.
21: **end for**
22: Update last route and last position of the nodes served in the current partial solution.
23: **if** *Pool* is empty **then**
24: Set *Iter* to 0.
25: Update best solution.
26: Initialize *Pool* with the nodes in the smallest route having the least number of nodes in the current solution. Remove the route from the solution.
27: **end if**
28: **end while**
29: **return** Best solution.

Figure 4.2. Large Neighborhood Search with Smart Node Mobility Heuristic Algorithm

4.2.1. Destroy Heuristic

Destroy heuristic is presented in Figure 4.3. It removes R many nodes from a given solution. If a route has only a node, the heuristic does not remove it to prevent the route becoming empty. It allows the algorithm to reduce the number of nodes one by one in the route reduction phase. Also, it guarantees that the total distance of a given solution is improved without reducing the number of tours. Therefore, *CandidateList* is initialized with the nodes which is not the single node in its route and the list is updated after each removal. Removal is performed one by one based on a roulette wheel selection. The list shrinks after each removal by one (or two when a node remains as the single node in a route after a removal), and the *Pool* grows. Roulette wheel selection probabilities are updated to take into account the composition of the Pool as well as the mobility tendencies of the candidate nodes as described in the next section.

1: Create lists as *CandidateList*.
2: **for each** node served in the current partial solution **do**
3: Add it to *CandidateList* if it is not the single node in its route.
4: **end for**
5: **while** the count of the removed nodes $<R$ **do**
6: *Update of the Roulette Wheel Selection Probabilities* of the nodes in *CandidateList*. Select a node randomly from *CandidateList* using roulette wheel selection.
7: Add the selected node to *Pool*, and remove it from *CandidateList*.
8: Remove the selected node from its route and update the route. Update last position values for the nodes in the route.
9: If there remains a single node in the route of the selected node, remove the single node from *CandidateList*.
10: **end while**
11: **return** *Pool*

Figure 4.3. Destroy Heuristic Algorithm

4.2.2. Update of the Roulette Wheel Selection Probabilities

Roulette wheel selection randomly selects a subset of candidates proportional to their fitness values. It is a useful tool to incorporate diversification via probabilistic selection and intensification via fitness value based probability magnitudes to a heuristic search mechanism. If fitness level of a candidate i is f_i, the probability of chosen i among all nodes in *CandidateList* during destroy heuristic is

$$p_i = \frac{f_i}{\sum_{j \in C} f_j} \tag{4.4}$$

In each iteration of LNS-SNM, the main objective is to remove a subset of nodes and serve all unserved customers in *Pool* to reduce the number of routes. Tendency value measures the willingness of a node to change its route or position. The higher tendency value is, the higher possibility to be inserted in a place is. On the other hand, *Pool* contains unserved nodes before destroyal. To place these nodes, LNS-SNM needs to remove closest nodes to *Pool* in terms of distance. If a distant node is removed, the probability of inserting it to the same place is higher. Using this two metrics, fitness function is defined as:

$$f_i = \frac{TV_i}{\frac{\sum_{j \in C} TV_j}{S}} * \frac{\frac{\sum_{j \in C} d_j}{S}}{d_i} \tag{4.5}$$

S is the size of the candidate list, TV_i is the tendency value of node i, d_i is the average linkage distance of all pairs between node i and the nodes in *Pool*. Fitness function is multiplication of the ratio of the tendency value of i to the average tendency value of all nodes in *CandidateList* and the inverse of the ratio of the distance value of i to the average distance value of all nodes in *CandidateList*.

4.2.3. Repair Heuristic

Repair heuristic tries to find feasible solutions by inserting the unserved nodes in *Pool*. *Basic greedy insertion* performs the cheapest insertion for all unserved nodes until all of them are inserted or no feasible insertion remains for the unserved nodes. *k-regret insertion* (Ropke and Pisinger, 2006), is somewhat less myopic than basic greedy heuristic since it performs the best insertion that we will regret more. In other word, the node having the largest sum of gaps between the first k best insertion cost and the best insertion cost. k can vary from 2 to the number of routes in a solution.

4.3. Distance Reduction

Distance reduction phase aims at improving the solution with a given number of routes. Small adjustments are made in LNS-SNM outlined in Figure 4.2 for distance improvement. *Pool* is only filled with removed nodes in each iteration. If *Pool* is not emptied LNS-SNM goes back to the last feasible solution. If *Pool* is empty, a feasible solution is obtained. The acceptance of the solution is evaluated using simulated annealing method. Acceptance probability is limited such that a solution can be at most $w\%$ worse than the previous. LNS-SNM updates *tendency* values for the removed nodes regarding the success of their insertions as before. Probabilities of the nodes for roulette wheel selection is based on only the magnitudes of their *tendency* values for the first node selection in destroy heuristic since the closeness is not calculated for an empty *Pool* in destroy heuristic. After the first removal, probabilities are updated as described in Update of the Wheel Roulette Selection Probabalities section which is a combination of *tendency* values and closeness to *Pool* metrics.

As well as the deviation parameter w, T for the initial temperature and c for the cooling rate of the simulating annealing are the parameters of the distance reduction phase in addition to γ, θ, β, α, $MaxIter$, $RepairHeuristic$ and R used in LNS-SNM. Figure 4.4 summarizes each step of LNS-SNM-D for distance reduction.

1: Input:α, β, θ, γ, MaxIter, R, $RepairHeuristic$, a feasible solution (x), w, c
2: Set $Iter$ =0
3: Create an empty list as $Pool$
4: Set $T = \frac{w*f(x)}{\ln \frac{1}{0.9}}$
5: **while** $Iter$<$MaxIter$ **do**
6: Increase $Iter$.
7: Set $T = T * c$.
8: Update last routes and last positions for all nodes.
9: Perform *Destroy(Pool,R)*.
10: Perform *Repair Heuristic* for the nodes in $Pool$. Keep a list for the nodes inserted as $InsertedNodes$.
11: **for each** Node in $Pool$ and $InsertedNodes$ **do**
12: Update the tendency value with appropriate parameter among α, β, θ, and γ.
13: **end for**
14: **if** $Pool$ is empty **then**
15: Name the obtained feasible solution as x'.
16: **if** $f(x') < f(x)$ **then**
17: Accept the solution and update best solution. Set $x = x'$.
18: **else if** $f(x') < (1 + w) * f(x)$ **then**
19: Generate a random number, $rndm$, from (0,1] interval.
20: **if** $rndm < e^{\frac{f(x)-f(x')}{T}}$ **then**
21: Accept the solution. Set $x = x'$.
22: **end if**
23: **else**
24: Do not accept the solution.
25: **end if**
26: **else**
27: Do not accept the solution.
28: Empty $Pool$.
29: **end if**
30: **end while**
31: **return** Best solution.

Figure 4.4. LNS-SNM-D

5. COMPUTATIONAL RESULTS

This chapter presents the extensive numerical studies conducted to assess the performance of the proposed heuristic LNS-SNM. First, set of experiments on different sets of problems taken from the literature on different variants of Vehicle Routing Problem with Backhauls (VRPB) verify the quality of our method. Then the computational results for the problem studied in this thesis, namely VRPMPDTW-BS, are provided. Finally, the method is used for the real life cases based on data provided by the LSP.

The proposed heuristic LNS-SNM is coded in C# and executed in a computer equipped with Intel® CoreTM i7 CPU @ 3.07 GHz processor and 8 GB RAM and Windows 10 64-bit operating system. The MILP models are solved using the callable library of IBM ILOG CPLEX Optimization Studio V12.10.0.

5.1. Parameter Tuning

As an initial step, we conduct experiments for fine tuning several parameters used by LNS-SNM. Ropke and Pisinger (2006) claim that destroying of 40% of all nodes gives better solutions in large neighborhood search methods. By following this inference, the number of removals, R is taken as 40% of the size of the problem in the reported results. Since different problem types tested have different side constraints, the initial values for mobility tendencies of nodes will be initialed differently in each problem type. But a general setting for tendency update parameters, γ, θ, β, and α, are searched over 0.990 to 1.000 interval by 0.001 increments on VRPMPDTW-BS test data with 50 nodes generated for this study. Repair heuristic is given as 2-regret and $MaxIter$ is used 800 for the route reduction and 2000 for the distance reduction phase during this parameter search. The best combination that gives the minimum average number of routes and the minimum total distance is found as 1.000, 0.998, 0.993 and 0.990 for γ, θ, β, and α, respectively. After deciding these parameters, $MaxIter$ is analyzed in increasing pattern from 500 to 2000 with a step size of 100 for route reduction and five times larger for the distance reduction. By comparing the increase in the

average quality of solution and the increase in computing time, it is taken as 1000 iterations for the route reduction phase and 5000 for the distance reduction phase separately. The number of iterations is way smaller than similar neighborhood searches in the literature due to high computations during *Update of the Roulette Wheel Selection Probabilities*. However, 5000 iterations are good enough to reduce the total distance significantly. Finally, parameter analysis is conducted for repair heuristics, namely basic greedy insertion, 2-regret, 3-regret and m-regret. 3-regret heuristic gives the best average solution first in the average number of routes then the total distance.

w and c for the simulating annealing procedure are set to 0.05 and 0.9998 as suggested by Ropke and Pisinger (2006) for the most of variants of vehicle routing problems with backhauls. T is calculated for each instance such that the probability of acceptance for a solution having 0.05 deviation in total distance from the total distance of the initial solution at the beginning of the distance minimization phase is equal to 0.90.

5.2. Assessment of the Proposed Heuristic on Other Problem Variants

In this section, the proposed heuristic is tested on different variants of Vehicle Routing Problems with Backhauls to compare the quality of the solutions with the existing approaches in the literature.

5.2.1. Vehicle Routing Problem with Backhauls (VRPB)

VRPB is a variant of CVRP where backhaul customers are served only after linehaul customers are visited. This means that a route may consist of only linehaul customers but a route cannot contain only backhauls customers. VRPB aims at finding the lowest total distance for a given number of vehicles, K.

CRBI starts with n separate and feasible tours where each customer is served by a different vehicle, then it merges the routes to reduce the number of routes. To generate initial feasible solutions to VRPB, CRBI requires modifications since a backhaul customer cannot

be served by a vehicle alone. To accommodate this additional constraint on route generation, CRBI generates as many routes as the number of linehaul customers. Then, backhaul customers are inserted into these routes using 3-regret insertion heuristic. If there exist unserved backhaul customers, they are kept as unserved customers. The route construction procedure continues by merging the routes and CRBI returns the solution when the number of constructed routes is equal to K to prevent constructing less routes which is possible for many cases. After generating initial routes for all customers except unserved backhaul customers, LNS-SNM is used by initializing the *Pool* with unserved backhaul customers. If it succeeds to empty the *Pool*, a feasible initial solution is generated.

Table 5.1 summarizes the results on 62 symmetric instances proposed by Goetschalckx and Jacobs-Blecha (1989). The number of customers varies between 25 and 150 customers. The first five columns show the name, the number of delivery customers, the number of pickup customers and the number of vehicles of each instance. The optimal solutions of all instances are proven by Queiroga et al. (2020) and the optimal distances are given under *Optimal Distance* column. The final five columns present the details of solutions generated by LNS-SNM. *NV* shows the number of vehicles. *Avg. Distance* is the average distance of solutions obtained in 10 replications of LNS-SNM while *Best Distance* is the minimum distance achieved in these replications. *Avg. Time* represents the average computing time of a single replication. *% Dev. of Best Distance* as given in Equation 5.1 is the deviation between the total distance of Best solution from the best known solution total distance. The row *Avg.* reports the average of optimal distances, average distances of 10 replications, best distances, computation times and deviations.

$$Dev = \frac{objective - best\ known\ objective}{best\ known\ objective} * 100 \tag{5.1}$$

In all instances of GJB, CRBI constructs as many routes as K. Hence, LNS-SNM for route reduction is not performed. LNS-SNM for distance reduction is carried out to improve the initial routes and tendency values for all nodes are initialized as 1 since there is no specific differences among nodes. The average gaps for successful heuristics in the literature is less than 0.5% while the best one is 0.02% according to the survey conducted by Koç and Laporte (2018). For the proposed heuristic, the average gap between the best solution and

the optimal solution is 0.25% while the average solution of 10 replications is 0.66 % away from the optimal solution on average. LNS-SNM produces competitive solutions for VRPB that validate the power of its search strategy as a distance minimization method by producing similar solutions compared to the methodologies in the literature.

Table 5.1. Results on VRPB test data proposed by Goetschalckx and Jacobs-Blecha (1989)

GJB Dataset						LNS-SNM				
Ins.	n+m	n	m	K	Optimal Distance	*NV*	Avg. Distance	Best Distance	Avg. Time (s)	%Dev. of Best Distance.
A1	25	20	5	8	229,885.65	8	229,885.65	229,885.65	1	0.00
A2	25	20	5	5	180,119.21	5	180,119.21	180,119.21	2	0.00
A3	25	20	5	4	163,405.38	4	165,214.93	163,405.38	3	0.00
A4	25	20	5	3	155,796.41	3	155,796.41	155,796.39	3	0.00
B1	30	20	10	7	239,080.16	7	239,080.16	239,080.16	2	0.00
B2	30	20	10	5	198,047.77	5	198,047.77	198,047.76	3	0.00
B3	30	20	10	3	169,372.29	3	169,372.29	169,372.29	5	0.00
C1	40	20	20	7	250,556.77	7	250,832.84	250,556.77	4	0.00
C2	40	20	20	5	215,020.23	5	215,020.23	215,020.23	6	0.00
C3	40	20	20	5	199,345.96	5	199,345.96	199,345.95	8	0.00
C4	40	20	20	4	195,366.63	4	195,366.63	195,366.61	8	0.00
D1	38	30	8	12	322,530.13	12	322,530.13	322,530.13	3	0.00
D2	38	30	8	11	316,708.86	11	316,755.07	316,708.86	3	0.00
D3	38	30	8	7	239,478.63	7	239,478.63	239,478.59	4	0.00
D4	38	30	8	5	205,831.94	5	205,925.11	205,831.94	6	0.00
E1	45	30	15	7	238,879.58	7	238,879.58	238,879.58	7	0.00
E2	45	30	15	4	212,263.11	4	212,746.22	212,263.11	11	0.00
E3	45	30	15	4	206,659.17	4	207,303.06	206,659.17	14	0.00
F1	60	30	30	6	263,173.96	6	264,922.01	263,620.77	14	0.17
F2	60	30	30	7	265,214.16	7	265,310.06	265,214.16	14	0.00
F3	60	30	30	5	241,120.78	5	241,249.89	241,120.78	21	0.00
F4	60	30	30	4	233,861.85	4	236,330.57	234,303.91	28	0.19
G1	57	45	12	10	306,305.40	10	306,756.23	306,305.40	9	0.00
G2	57	45	12	6	245,440.99	6	245,440.99	245,440.99	16	0.00
G3	57	45	12	5	229,507.48	5	229,507.48	229,507.48	20	0.00
G4	57	45	12	6	232,521.25	6	232,925.42	232,521.25	20	0.00
G5	57	45	12	5	221,730.35	5	221,794.53	221,730.35	25	0.00

Table 5.1. Results on VRPB test data proposed by Goetschalckx and Jacobs-Blecha (1989) (cont.)

GJB Dataset						LNS-SNM				
Ins.	n+m	n	m	K	Optimal Distance	*NV*	Avg. Distance	Best Distance	Avg. Time (s)	%Dev. of Best Distance.
G6	57	45	12	4	213,457.45	4	217,522.04	213,457.45	34	0.00
H1	68	45	23	6	268,933.06	6	269,601.56	268,958.53	28	0.01
H2	68	45	23	5	253,365.50	5	253,721.43	253,365.50	36	0.00
H3	68	45	23	4	247,449.04	4	248,622.51	247,449.04	45	0.00
H4	68	45	23	5	250,220.77	5	251,109.12	250,220.77	44	0.00
H5	68	45	23	4	246,121.31	4	248,892.21	246,121.31	51	0.00
H6	68	45	23	5	249,135.32	5	251,004.67	249,279.88	49	0.06
I1	90	45	45	10	350,245.28	10	351,079.60	350,435.13	32	0.05
I2	90	45	45	7	309,943.84	7	312,594.03	309,943.84	45	0.00
I3	90	45	45	5	294,507.38	5	296,239.83	294,868.24	73	0.12
I4	90	45	45	6	295,988.45	6	297,181.75	295,988.45	63	0.00
I5	90	45	45	7	301,236.01	7	303,480.26	301,739.39	59	0.17
J1	94	75	19	10	335,006.68	10	336,050.13	335,356.50	43	0.10
J2	94	75	19	8	310,417.21	8	312,117.00	310,793.35	58	0.12
J3	94	75	19	6	279,219.21	6	282,719.55	279,219.21	91	0.00
J4	94	75	19	7	296,533.16	7	297,431.83	296,800.20	71	0.09
K1	113	75	38	10	394,071.17	10	399,659.93	397,387.97	71	0.84
K2	113	75	38	8	362,130.00	8	366,281.99	363,276.47	95	0.32
K3	113	75	38	9	365,694.08	9	369,921.56	368,543.85	95	0.78
K4	113	75	38	7	348,949.39	7	351,952.56	348,949.39	122	0.00
L1	150	75	75	10	417,896.71	10	428,541.62	426,133.78	156	1.97
L2	150	75	75	8	401,228.80	8	403,098.91	401,375.69	215	0.04
L3	150	75	75	9	402,677.72	9	405,816.99	402,677.72	209	0.00
L4	150	75	75	7	384,636.33	7	389,736.20	385,885.77	274	0.32
L5	150	75	75	8	387,564.55	8	394,252.86	390,489.68	266	0.75
M1	125	100	25	11	398,593.19	11	406,555.74	402,798.54	106	1.06
M2	125	100	25	10	396,916.97	10	403,526.77	400,312.58	106	0.86
M3	125	100	25	9	373,010.93	9	378,255.23	376,817.98	128	1.02
M4	125	100	25	7	348,140.16	7	350,047.81	348,719.50	181	0.17
N1	150	100	50	11	408,100.62	11	416,668.67	412,559.70	191	1.09
N2	150	100	50	10	408,065.44	10	417,299.87	411,024.73	191	0.73
N3	150	100	50	9	394,337.86	9	402,186.93	397,446.35	226	0.79

Table 5.1. Results on VRPB test data proposed by Goetschalckx and Jacobs-Blecha (1989) (cont.)

GJB Dataset						LNS-SNM				
Ins.	n+m	n	m	K	Optimal Distance	*NV*	Avg. Distance	Best Distance	Avg. Time (s)	%Dev. of Best Distance.
N4	150	100	50	10	394,788.36	10	401,586.41	400,198.71	204	1.37
N5	150	100	50	7	373,476.30	7	377,578.42	375,561.66	299	0.56
N6	150	100	50	8	373,758.65	8	380,954.17	379,795.12	295	1.62
Avg.					290,532.92		292,890.76	291,484.92	72.74	0.25

5.2.2. Vehicle Routing Problem with Backhauls and Time Windows (VRPBTW)

VRBTW requires backhaul customers to be served after linehaul customers if there is any linehaul customer in their routes. In other words, it allows routes having only backhaul customers unlike VRPB but still first linehaul then backhaul constraint exists for the routes serving both linehauls and backhauls. Additionally, the size of the fleet is not fixed and the first priority in the objective is given to minimizing the number of vehicles over minimizing the total distance. CRBI can an generate initial feasible solution without any modification as outlined in Figure 4.1. The tendency values of nodes are set to the difference between the latest service time and the earliest service time. It is assumed that the longer the time window duration of a customer is, the more ability to relocate in a different place.

Table 5.2 summarizes the results on Gélinas et al. (1995) test instances. For comparison, methodologies that outperform other heuristics in terms of the average number of vehicles in the literature are presented. RU stands for Reimann and Ulrich (2006), RP stands for Ropke and Pisinger (2006), and TAP stands for Tarantilis et al. (2013). Optimalities of all instances are proven by Queiroga et al. (2020) and the optimal number of vehicles and distances are given under *Optimal* column with *NV* and *Distance* columns. For each study mentioned above, *NV* for the number of vehicles and *Distance* for the total distance of the solutions reported by the authors are given. LNS-SNM heuristic reduces the number of the vehicles in

the initial solution up to five vehicles and it produces the optimal number of vehicles for all instances except *BHR104A* and *BHR104B* which are more difficult to solve as can be seen in the solution time required for the optimal solution and the best existing heuristics. Only TAP reduced *BHR104A* instance to 10 vehicles. The average deviation between the best distance found in 10 replications and the optimal distance is -0.94% while the maximum deviation is 2.14%. Table 5.2 shows that LNS-SNM compares favorably with existing heuristics in terms of the number of vehicles and the total distance while TAP is slightly better than all of the methods.

5.2.3. Vehicle Routing Problem with Mixed Pickups and Deliveries, and Time Windows (VRPMPDTW)

Table 5.3 provides the results on the test instances generated by Kontoravdis and Bard (1995). There are few studies in the literature for vehicle routing problem with mixed backhauls and time windows. VRPMPDTW allows serving pickup customers and delivery customers in any order subject to time windows. RP stands for the best solutions obtained by the method of unified large neighborhood search of Røpke and Pisinger (2004), ZC for the best solutions produced by the method of guided local search of Zhong and Cole (2005), and TAP for the best solutions generated by the method of adaptive relinking path method of Tarantilis et al. (2013).

Tendency values are initialized in the same manner as used in VRPBTW, as the size of time windows of nodes. LNS-SNM is successful at minimizing the number of vehicles for all test instances in Kontoravdis and Bard (1995) data set while maintaining average percent deviation of 0.13% for MC2, 1.5% for MR2 and 3.03% for MRC2 separately. The performance of LNS-SNM is relatively worse than RP and TAP solutions in terms of total distance but still produces solutions within at most 4.67 % deviation from the best known solution. The average computation time is 367.82 seconds for RP, 437.00 seconds for ZC, 62.91 seconds for TAP and 157.54 for LNS-SNM.

Table 5.2. Results on VRPBTW test instances generated by Gélinas et al. (1995)

	Optimal				RP				RU				TAP				LNS-SNM					
Instance	% of Backhaul Customers	*NV*	Distance	Avg. Time (s)	NV	Distance	Avg. Time (s)	Gap %	*NV*	Distance	Avg. Time (s)	Gap %	*NV*	Distance	Avg. Time (s)	Gap %	Avg. *NV*	Avg. Distance	Best *NV*	Best Distance	Avg. Time (s)	% Dev. of Best Distance
BHR101A	10	22	1818.86	1	22	1818.86	109	0.00	22	1818.86	-	0.00	22	1818.86	16	0.00	22	1821.14	22	1818.86	20.71	0.00
BHR101B	30	23	1959.52	1	23	1959.56	103	0.00	23	1959.56	-	0.00	23	1959.52	68	0.00	23.1	1958.11	23	1959.67	20.78	0.01
BHR101C	50	24	1939.1	1	24	1939.1	101	0.00	24	1939.1	-	0.00	24	1939.1	75	0.00	24	1939.83	24	1939.10	21.28	0.00
BHR102A	10	19	1653.18	2	19	1653.19	121	0.00	19	1653.19	-	0.00	19	1653.18	16	0.00	19	1653.51	19	1653.18	33.90	0.00
BHR102B	30	22	1750.7	1	22	1750.7	114	0.00	22	1750.7	-	0.00	22	1752.28	150	0.09	22	1750.71	22	1750.70	28.68	0.00
BHR102C	50	22	1775.76	1	22	1775.76	113	0.00	22	1775.76	-	0.00	22	1775.76	25	0.00	22	1776.94	22	1775.76	29.57	0.00
BHR103A	10	15	1385.38	2	15	1387.57	128	0.16	15	1387.57	-	0.16	15	1385.38	24	0.00	15	1390.74	15	1387.51	39.46	0.15
BHR103B	30	15	1390.32	3	15	1390.33	115	0.00	15	1390.33	-	0.00	15	1390.32	17	0.00	15	1399.95	15	1390.97	32.80	0.05
BHR103C	50	17	1456.48	2	17	1456.48	115	0.00	17	1456.48	-	0.00	17	1456.48	73	0.00	17	1459.81	17	1456.48	34.15	0.00
BHR104A	10	10	1202.53	1437	11	1084.17	132	-9.84	11	1084.17	-	-9.84	10	1203.44	195	0.08	11	1100.62	11	1087.47	48.70	-9.57
BHR104B	30	10	1258.48	55	11	1154.84	122	-8.24	11	1154.84	-	-8.24	11	1154.84	209	-8.24	11	1183.94	11	1162.79	38.66	-7.60
BHR104C	50	11	1188.78	11	11	1191.38	119	0.22	11	1191.38	-	0.22	11	1194.73	203	0.50	11	1224.19	11	1214.24	38.34	2.14
BHR105A	10	15	1560.15	86	15	1561.28	109	0.07	15	1561.28	-	0.07	15	1560.15	151	0.00	15.5	1531.69	15	1570.49	20.38	0.66
BHR105B	30	16	1583.3	1	16	1583.3	102	0.00	16	1583.3	-	0.00	16	1583.3	1	0.00	16	1592.15	16	1583.30	19.78	0.00
BHR105C	50	16	1709.66	51	16	1710.19	100	0.03	16	1710.19	-	0.03	16	1709.66	54	0.00	16.5	1636.18	16	1710.75	20.01	0.06
Average					17.27	1561.11	113.53	-1.17	17.27	1561.11	141.00	-1.17	17.20	1569.13	85.13	-0.50	17.34	1561.30	17.27	1564.08	29.81	-0.94

Table 5.3. Results on VRPMPDTW test instances proposed by Kontoravdis and Bard (1995)

	Best Known Solution			RP			ZC			TAP			LH					
	Ins.	NV	Distance	NV	Distance	% Dev.	NV	Distance	% Dev.	NV	Distance	% Dev.	Avg. NV	Avg. Distance	Best NV	Best Distance	Avg. Time (s)	% Dev. Of Best Distance
MC2	mc201	4	766.82	4	766.82	0.00	5	763.88	-0.38	-	-	-	4	807.04	4	766.82	27.51	0.00
	mc202	4	732.93	4	732.93	0.00	4	1186.24	61.85	-	-	-	4	747.76	4	732.93	113.61	0.00
	mc203	4	704.49	4	704.49	0.00	4	1096.31	55.62	-	-	-	4	711.24	4	704.49	164.51	0.00
	mc204	4	676.18	4	676.18	0.00	4	885.73	30.99	-	-	-	4	697.16	4	677.69	292.24	0.22
	mc205	4	748.34	4	748.34	0.00	5	781.7	4.46	-	-	-	4	763.4	4	751.96	47.10	0.48
	mc206	4	747.08	4	747.08	0.00	5	860.74	15.21	-	-	-	4	768.98	4	749.85	65.05	0.37
	mc207	4	737.39	4	737.39	0.00	5	792.96	7.54	-	-	-	4	765.11	4	737.39	69.66	0.00
	mc208	4	735.17	4	735.17	0.00	5	859.92	16.97	-	-	-	4	747.15	4	735.17	74.27	0.00
	Avg.	4	731.05	4	731.38	0.00	4.625	903.56	24.03	-	-	-	4	750.98	4	732.04	106.74	0.13
MR2	mr201	4	1236.31	4	1256.31	1.62	4	1388.73	12.33	4	1236.31	0.00	4	1298.03	4	1277.94	45.63	3.37
	mr202	4	1086.46	4	1086.46	0.00	4	1198.99	10.36	4	1086.46	0.00	4	1112.81	4	1093.40	137.69	0.64
	mr203	4	894.54	4	894.54	0.00	4	988.82	10.54	4	894.54	0.00	4	931.72	4	914.87	273.47	2.27
	mr204	4	736.75	4	736.75	0.00	4	858.32	16.50	4	737.43	0.09	4	771.75	4	741.41	328.17	0.63
	mr205	4	974.26	4	974.26	0.00	4	1172.53	20.35	4	974.26	0.00	4	1014.38	4	992.58	127.86	1.88
	mr206	4	893.67	4	894.04	0.04	4	979.5	9.60	4	893.67	0.00	4	939.27	4	894.04	228.01	0.04
	mr207	4	800.79	4	800.79	0.00	4	912.69	13.97	4	800.79	0.00	4	849.24	4	819.37	295.92	2.32
	mr208	4	716.28	4	716.28	0.00	4	764.52	6.73	4	716.28	0.00	4	739.7	4	725.37	330.19	1.27
	mr209	4	877.04	4	879.63	0.30	4	978.82	11.60	4	877.04	0.00	4	910.83	4	892.37	191.62	1.75
	mr210	4	924.56	4	924.56	0.00	4	1061.36	14.80	4	924.56	0.00	4	955.88	4	938.05	209.33	1.46
	mr211	4	763.09	4	765.03	0.25	4	878.81	15.16	4	763.09	0.00	4	792.67	4	769.33	269.74	0.82
	Avg.	4	900.34	4	902.73	0.20	4	1016.66	12.91	4	900.40	0.01	4	937.84	4	914.43	221.60	1.50
MRC2	mrc201	5	1338.96	5	1355.63	1.24	5	1498.9	11.95	5	1338.96	0.00	5	1408.07	5	1366.87	41.50	2.08
	mrc202	4	1223.7	4	1230.24	0.53	4	1539.41	25.80	4	1223.7	0.00	4	1292.35	4	1280.87	125.36	4.67
	mrc203	4	987.8	4	995.63	0.79	4	1303.48	31.96	4	987.8	0.00	4	1039.24	4	1022.83	220.32	3.55
	mrc204	4	833.6	4	836.89	0.39	4	932.48	11.86	4	833.6	0.00	4	865.02	4	846.63	205.06	1.56
	mrc205	4	1411.19	4	1414.52	0.24	4	1632.04	15.65	4	1411.19	0.00	4	1500.98	4	1458.72	84.28	3.37
	mrc206	4	1221.74	4	1254.51	2.68	4	1433.43	17.33	4	1221.74	0.00	4	1311.28	4	1273.07	88.91	4.20
	mrc207	4	1066.24	4	1083.33	1.60	4	1217.2	14.16	4	1066.24	0.00	4	1143.22	4	1098.84	138.03	3.06
	mrc208	4	843.58	4	849.3	0.68	4	1085.57	28.69	4	843.58	0.00	4	895.9	4	858.48	250.68	1.77
	Avg.	4.13	1115.85	4.13	1127.51	1.02	4.13	1330.31	19.67	4.13	1115.85	0.00	4.13	1182.01	4.13	1150.79	144.27	3.03

5.3. Vehicle Routing Problem with Mixed Pickup and Deliveries, Time Windows, Shifts and Meal Breaks (VRPMPDTW-BS)

Since VRPMPDTW-BS explored in this thesis has not been studied before in the literature, there is no benchmark instance to assess the quality of the solutions produced by our heuristic. Therefore, we study on two instance sets. The first set is composed of smaller instances which are solved using the mathematical models formulated in Chapter 3 for assessing the quality of the solutions obtained by the LNS-SNM. The second set of larger test instances is provided by the LSP to compare the running times and solution quality with respect to another method based on cluster-first, route-second approach.

5.3.1. Generation of Data Set

We generate several test instances from Solomon (1987) VRPTW benchmark instances. These test instances are classified as C1, C2, R1, R2, RC1 and RC2 and each contains 100 customers. In C1 and C2 data sets, locations of customers are generated as clustered Cartesian points while locations of customers are randomly generated from a uniform distribution in R1 and R2 sets. In RC1 and RC2, locations of customers are a mix of clustered and randomly distributed Cartesian points. In R1, C1, and RC1, customers have tight time windows, and vehicles have shorter scheduling horizons and smaller capacities allowing fewer number of customers in routes. In contrast, C2, R2 and RC2 data sets contain vehicles having longer scheduling horizons and larger capacities which result in many customers served in a single route.

In this study, we use C2 data set for generating our test instances for VRPMPDTW-BS. The reasons for choosing C2 data set are threefold. First, the distribution of customer locations provided by the logistics company is clustered. Second, vehicles that are identical have larger capacities, and third longer scheduling horizons.

Instances in C2 data set, route duration is limited up to 3390 units of time. To convert these instances to VRPMPDTW-BS instances, shift starting and ending times, and break

duration, starting and ending times should be specified. They are determined proportional to the data provided by the logistic company. The company can serve customers between 8:15 am and 22:15 pm, which is 840 minutes. The first shift takes place from 8:15 am to 15:45 pm while the second shift operates from 14:45 pm to 22:15 pm. The first shift constitutes the first 53.6% of the operation horizon and the second shift makes the last 53.6% of the operation horizon. Both shifts can serve between 14:45 pm and 15:45 pm which represents 7.2% of the operation horizon. Regarding this partitioning of the operation horizon, scheduling horizon of the first shift is set from 0 to 1817 time units while the scheduling horizon of the second shift is from 1573 to 3390. Using the same proportionality, the meal break for the first shift takes places between 909 and 1031 time units. It is between 2482 and 2604. The break duration is 122 time units. Service duration is set as 28 time units for deliveries and 40 time units for pickups in these test instances generated for VRPMPDTW-BS, in proportion to 7 minutes and 10 minutes service durations given by our benchmark logistic company.

Vehicles should not perform any type of service during meal breaks. This makes C201, C202, C203 and C204 sets infeasible, since time windows of some of the customers coincide with meal break periods. Hence these instances are not used. For the remaining, first 25, 50 and 100 customers are taken to generate different size of instances. Additionally, for 33 % backhauling, backhaul customers are selected as customers with index three, six, nine and so on. For 50% backhauling, the same manner is adopted as customers are chosen as backhaul customers with index two, four, six and so on. 24 different test instances are designed for three different sizes and two different backhaul percentages for each test data C205, C206, C207 and C208.

5.3.2. Computational Results on Randomly Generated Test Instances

Table 5.4 summarizes the solutions of the exact method and the proposed heuristic on the randomly generated test instances. The first six columns under *Instances* block specify each instance, respectively the name of Solomon (1987) instance, the size of the instance, the percentage of the pickups, Type 0, Type 1 and Type 2 customers. *MILP Resulst* provides the optimal or best known solution to each instance. As mentioned in Chapter 3, each instance

is first solved for minimizing the number of vehicles for six hours. Then, the exact model is used for optimizing the total distance given the best number of vehicles found in the first phase as an upper bound on the number of vehicles for another six hours. *LB on NV* is the lower bound on the number of vehicles obtained by the solver during the first phase. *Best Known NV* is the number of vehicles and *Best Known Distance* is the total distance of the best solution found in the distance improvement phase. The instances with 25 nodes are solved to optimality. *LNS-SNM* block reports the performance of the proposed heuristic on these test instances. For these problems, tendency values are initialized according to types of nodes and the width of their time windows. If a node can be only served in a specific shift, Type1 or Type2, the tendency of the node is the width of its time window. If it can be served in both shifts, Type0, its tendency value is the twice of its interval size of the time window. It means that Type0 customers do not have shift incompatibility and have more ability to be relocated in other routes. Type1 and Type2 customers can be replaced only in a smaller subset of vehicles to prevent shift incompatibility.

The proposed heuristic is successful at minimizing the number of vehicles which has priority over reducing the total distance for VRPMPDTW-BS. It achieves the lower bound on the number of vehicles obtained by the exact method for 18 out of 24 instances. For the cases where LNS-SNM obtains less number of vehicles than MILP solutions, comparison with respect to total distance could be misleading since smaller number of vehicles does not guarantee lower total distance as can be seen from the solution of the instance *C207* with 50 nodes and 33% pickup customers. On the other hand, when the best solution generated by LNS-SNM has the same number of vehicles with the best known MILP solution, the deviation is at most 0.71 %. The average deviation for such cases, 14 out of 24 instances, is -0.11 %. Therefore, the proposed heuristic is a powerful search method for minimizing both the number of vehicles and the total distance for VRPMPDTW-BS.

Appendix A and Appendix B show the optimal tours produced by MILP for the instance C206 with 25 customers and 50% pickups customers. It verifies the proposed mathematical formulation in terms of serving the customers in a mixed order, meal breaks, shifts and capacity constraints as well as the time windows limitations.

Table 5.4. Results on the test data generated from Solomon (1987) instances

Instance						MILP Results					LNS-SNM					
Ins.	N	% of Pickup Customers	% of Type 0 Customers	% of Type 1 Customers	% of Type 2 Customers	LB on *NV*	Best Known *NV*	Best Known Distance	Gap (%)	Run Time (s)	Avg. *NV*	Avg. Distance	Best *NV*	Best Distance	Avg. Time (s)	% Dev. of Best Distance
C205	25	33	12	40	48	2	2	291.63	0.00	3.25	2.0	294.17	2	291.63	14.47	0.00
C206	25	33	16	40	44	2	2	270.60	0.00	9.11	2.0	270.60	2	270.6	17.71	0.00
C207	25	33	24	40	36	2	2	261.83	0.00	1611.23	2.0	261.83	2	261.83	22.93	0.00
C208	25	33	24	36	40	2	2	239.04	0.00	15.92	2.0	239.04	2	239.04	22.36	0.00
C205	25	50	12	40	48	2	2	293.82	0.00	3.50	2.0	329.10	2	293.82	13.62	0.00
C206	25	50	16	40	44	2	2	266.68	0.00	7.20	2.0	266.68	2	266.68	17.02	0.00
C207	25	50	24	40	36	2	2	257.15	0.00	1492.26	2.0	257.15	2	257.15	23.35	0.00
C208	25	50	24	36	40	2	2	239.95	0.00	37.15	2.0	239.95	2	239.95	22.49	0.00
C205	50	33	14	44	42	3	4	516.82	0.00	4860.10	4.0	521.42	4	516.82	54.68	0.00
C206	50	33	20	42	38	3	4	477.75	7.86	21697.55	4.0	481.40	4	481.14	64.74	0.71
C207	50	33	34	36	30	3	4	463.17	13.04	21608.31	3.0	533.61	3	532.37	99.04	14.94
C208	50	33	26	40	34	3	4	479.60	15.67	21610.13	4.0	476.25	4	476.08	55.69	-0.73
C205	50	50	14	44	42	3	4	503.43	0.86	21604.18	4.0	506.91	4	503.43	33.08	0.00
C206	50	50	20	42	38	3	4	462.62	9.35	21601.48	4.0	463.95	4	463.95	51.62	0.29
C207	50	50	34	36	30	3	4	494.12	20.20	21606.29	3.8	477.55	3	538.88	79.71	9.06
C208	50	50	26	40	34	3	4	462.53	13.52	21609.23	4.0	454.60	4	454.42	64.27	-1.75
C205	100	33	18	45	37	6	7	840.28	11.60	21603.54	6.0	916.29	6	903.64	187.74	7.54
C206	100	33	23	43	34	6	7	825.66	17.62	21611.97	6.0	869.86	6	863.61	246.15	4.60
C207	100	33	29	40	31	6	8	849.96	20.01	21612.04	6.0	873.29	6	855.58	293.76	0.66
C208	100	33	27	42	31	6	7	832.08	20.15	21602.95	6.0	847.24	6	824.39	310.02	-0.92
C205	100	50	18	45	37	5	6	794.40	8.63	21602.00	5.9	855.38	5	797.66	206.16	0.41
C206	100	50	23	43	34	5	6	779.46	13.95	21602.77	5.0	971.58	5	959.8	295.10	23.14
C207	100	50	29	40	31	5	7	1019.25	35.42	21603.52	5.0	968.04	5	897.86	358.81	-11.91
C208	100	50	27	42	31	5	7	850.55	24.85	21603.92	5.0	906.37	5	885.69	375.87	4.13

5.3.3. Real-life Application

In order to evaluate the quality of the solutions generated by our heuristic, we implement a method that is inspired by the traditional method employed by the company. The company used to divide its customers into clusters with respect to predefined geographic regions and determine a single route for each cluster using a variant of the nearest insertion heuristic. This was possible as the number of customers in each cluster was small enough to be served by a vehicle. However, this method cannot be used as a benchmark since it does not consider meal breaks, shifts and time windows constraints. Therefore, we extend it by incorporating dynamic clustering into our heuristic. This means that the customers in the original problem instance are partitioned into a number of clusters by using k-means clustering method first proposed by MacQueen (1967) based on geographic closeness and our heuristic is applied to each of those clusters, separately.

The LSP operates in Istanbul. European and Asian sides have their own depots and they are independent from each other. Therefore, two separate problems are solved in daily basis. To test the proposed heuristic for large real-life instances, we work on daily problems for each side in Istanbul provided by LSP. These instances vary from 956 to 1550 customers. For each instance, the nodes are clustered for a given cluster size k and LNS-SNM is applied independently to each cluster. The assumption of independence between the clusters enables us to employ parallel computing to solve VRPMPDTW-BS for each clusters. Therefore, the computation time to generate a complete solution to each instance is the largest computation time of the solution times of all clusters. Stopping condition is either to perform three replications using the maximum number of iterations explained in Section 5.1 or to hit 45 minutes time limit which is operationally allowed by the LSP. The algorithm is allowed to search for route reduction at most 20 minutes and stop the distance reduction if the total solution time exceeds 45 minutes time limit. Dividing the time limit enables us to search for improved solutions in both the number of vehicles and the total distance. For clusters having larger number of customers than 500, the algorithm might terminate before performing the predefined number of iterations due to the time limit. Reported computing times are the required time to perform three replications or the time limit for the maximum sized cluster.

The final solution is the sum of the solutions of k clusters.

In Table 5.5 we present the results of different number of clusters used in k-means clustering in order to see its effect on the performance. The first column of the table indicates the operation day and the side. The second column shows the number of customers to visit in that instance. For each k value, *Max Clust. Size* gives the maximum number of nodes among k clusters. *NV* and *Distance* represents respectively the total number of vehicles and the total distance to serve all nodes in each instance, which are the sums of the number of vehicles and the total distance values of k clusters. *Max Run Time (min)* is the solution time of the maximum sized cluster in minutes. The first observation is that that dividing the customers into eight or ten clusters result in more routes in all of the test instances. There is no obvious relationship between the number of clusters and the number of vehicles for the cases having less than eight clusters. One might think that the less the number of clusters is, the less the number of vehicles is but due to the time limit, the algorithm stops before searching for further improvements in the number of vehicles and the total distance as can be seen from the *Max Run Time (min)*. Smaller number of clusters means that the largest clusters might grow especially if it is less than six clusters. The computing time exponentially increases with respect to the size of the instance due to long widths of time windows, the number of Type 0 customers, the closeness of the customer locations and mixed pickup delivery option. These factors increase the feasible space for the insertions of unserved customers during the removal and insertion procedures. Another reason might be that the k-means clustering does not consider temporal information of customers and it might yield inefficient clustering for these instances.

Table 5.5. Clustering results on large scale real-life instances

Instance		No Cluster			$k = 2$				$k = 4$				$k = 6$				$k = 8$				$k = 10$			
Day - Side	Size	*NV*	Distance	Max Run Time (min)	Max Clust. Size	*NV*	Distance	Max Run Time (min)	Max Clust. Size	*NV*	Distance	Max Run Time (min)	Max Clust. Size	*NV*	Distance	Max Run Time (min)	Max Clust. Size	*NV*	Distance	Max Run Time (min)	Max Clust. Size	*NV*	Distance	Max Run Time (min)
1 - Asian	1315	43	4589.80	45.02	952	43	4280.93	45.01	658	**43**	**4127.73**	45.60	390	44	3989.67	30.62	260	45	4035.84	14.50	218	47	4049.24	9.32
1 - European	1550	52	6312.19	45.03	1185	**51**	**6351.41**	45.01	695	53	5487.34	45.89	427	52	5351.48	46.80	376	53	5358.96	43.83	371	56	5399.60	42.19
2 - Asian	1120	40	3910.12	45.01	797	40	3717.72	45.01	585	39	3679.89	46.70	284	**38**	**3626.81**	16.97	260	39	3625.71	13.40	188	39	3601.16	5.97
2 - European	1412	48	5859.94	45.02	942	**48**	**5235.80**	45.01	593	50	5133.13	45.87	431	49	5210.55	41.73	337	50	5057.85	31.52	314	51	5093.68	31.60
3 - Asian	1007	35	3496.21	45.00	736	**35**	**3267.64**	45.01	426	35	3271.66	35.39	315	35	3273.95	18.89	217	36	3280.12	9.33	190	38	3258.94	8.16
3 - European	1205	42	4604.71	45.00	734	**41**	**4381.73**	45.00	555	43	4464.16	46.30	328	43	4342.00	26.05	280	44	4401.79	25.95	220	47	4422.83	13.43
4 - Asian	926	33	3341.15	45.01	665	32	3208.64	45.00	489	**31**	**3256.07**	41.33	237	32	3100.09	9.71	198	36	3126.70	6.84	165	36	3114.98	6.17
4 - European	1206	**41**	**4633.84**	45.01	754	42	4347.45	45.01	531	44	4432.25	44.35	358	44	4309.25	27.58	223	46	4339.74	10.47	284	47	4405.51	22.32

Table 5.6. Comparison of clustering results

Number of Clust.	Avg. NV	Avg. Cost	Avg. Time (min)	Diff. in Avg. NV from Base Scen.	% Dev. From from Base Scen.	% Dev. In Run Time (min) from Base Scen.	Number of Best Solutions out of 8 Instances
Base Scenario (No Cluster)	41.75	4593.49	45.01	-	-	-	1
2	41.50	4348.91	45.01	-0.25	-5.32%	-0.01%	4
4	42.25	4231.53	43.93	0.50	-7.88%	-2.41%	2
6	42.13	4150.47	27.29	0.38	-9.64%	-39.37%	1
8	43.63	4153.34	19.48	1.88	-9.58%	-56.73%	0
10	45.13	4168.24	17.39	3.38	-9.26%	-61.36%	0

Table 5.6 summarizes the average number of vehicles, distances and the computation times of eight instances for each clustered or non-clustered scenario reported in Table 5.5. For comparison, the scenario without clustering (no cluster) is chosen as a base case scenario to see the advantages and disadvantages of clustering. For each k, *Diff. in Avg. NV from Base Scen.* is the increase in the number of average vehicles and *% Dev. in Avg. Distance from Base Scen.* is the percent increase in the average distance with respect to the base scenario. The seventh column reports the percent increase in the average computation time for each case. The final column gives the number of instances, out of 8 instances, for which the corresponding case obtains the best solution The best option in terms of the number of vehicles is to divide the customers into two clusters which also improves the total distances with respect to no clustering option. It solves half of the instances better than other cases. However, partitioning into six clusters increases the average number of clusters by 0.38 but improves the average total distance by 9.64% while also reducing the computation time by 39.37%.

The LSP currently does not provide time window options to its customers but will launch it, soon. Therefore, there is no way to compare our methodology with any solution provided by the LSP. However, they favor the proposed heuristic since it reduces the number of vehicles below the size of the fleet available in each depot to serve customers as well as the average distance traveled by a vehicle.

6. CONCLUSION

In this study we address an extension of the classical Vehicle Routing Problem with Mixed Pickup, Delivery and Time Windows, which incorporates shifts as well as meal breaks of the personnel. We develop three methods for the solution of this problem which we name as the Mixed Pickup and Delivery Vehicle Routing Problem with Time Windows, Meal Breaks and Shifts (VRPMPDTW-BS) we develop three methods. The first method involves a mixed-integer linear programming model that can be solved by commercial solvers such as CPLEX and Gurobi to optimality for the instances up to 25 nodes. Note that there do not exist published works in the literature which report feasible solutions for Vehicle Routing Problem with Time Windows including lunch breaks for 25 nodes.

The second method is based on a three-phase heuristic to solve large VRPMPDTW-BS instances. The first phase of the solution approach consists of a construction heuristic that generates a feasible solution with respect to all the restrictions imposed. This solution is improved in terms of the number of vehicles in the second phase. For route minimization, a large neighborhood search is designed called as Large Neighborhood Search with Smart Node Mobility (LNS-SNM) that utilizes the information of consecutive removal and insertion moves to learn the best positions of nodes in the solution. In contrast to the similar studies in the literature that use multiple destroy and repair heuristics in an adaptive way, LNS-SNM uses a single destroy and repair heuristic to show the power of node mobility as a guidance in neighborhood search for route and distance minimization. Tendency values as a measure to monitor the nodes are used in the destroy phase. For the last phase, distance minimization, LNS-SNM is hybridized with simulating annealing to improve total distance for a given number of vehicles. The proposed heuristic is tested on different variants of Vehicle Routing Problem with Backhauls. The results are comparable to the best existing heuristics in terms of the fleet size, total distance and the computing time. Additionally, small-sized instances for VRPMPDTW-BS are created and solved by the exact method optimally for 25 nodes. The solutions provided by the exact method is a good guidance to assess the quality of the proposed heuristic even if the optimality is not proven for relatively larger instances with 50

or 100 nodes. The lower bounds for the number of vehicles for these instances obtained by the mathematical model are achieved for 18 out of 24 instances. Also, the average percent deviation is 2.09%.

The third method is to partition the customers into non-overlapping clusters in terms of the geographical locations by means of the well-known k-means clustering algorithm, and then applying our methodology to the customer set of each cluster. Obviously, the third approach is the fastest method, as the number of customers becomes smaller in each cluster. But this advantage comes at the expense of reduced solution quality. From the observations on large real-life instances, partitioning into six clusters is more advantageous in terms of the tradeoff regarding the computing time to the solution quality. Also, more clusters result in more routes due to ignoring the temporal information of customers in k-means clustering that can cluster customers with different time window requirements into smaller clusters and yields more routes for these smaller subsets.

There exist several directions in which this study can be further extended. It could be interesting to devote some research effort for the investigation of partitioning customers into clusters taking into account not only spatial but also temporal characteristics of the customers. Clustering becomes a nontrivial problem in the presence of time windows and especially when there exist shifts because the distance measure between the customer nodes should consider their spatial and temporal attributes. In other words, closeness of the customers must be defined with respect to both geographic position as well as time windows even though most distance measures are based on geographic information. The problem specified in this study adds another dimension, which is the shift information, making the clustering even more difficult. Another challenging extension involves the dynamic re-construction of the routes under real-time modifications in the specifications of the tasks during the daily operations. These modifications may include the cancellations of the pickup or delivery requests, new requests, changes in the addresses or time windows. This kind of real-time problem is another operational problem of the LSP and it could be challenging to adapt approaches developed in this thesis to design decision-making algorithms to reduce the daily operational costs to respond the requests of the customers. Price discrimination topic could be an interesting part

of this problem since the time window options can be priced with respect to how easy and cheap to serve the customer request in these options at any point of the time depending on the availability within the routes.

REFERENCES

Belmecheri, F., C. Prins, F. Yalaoui and L. Amodeo, "Particle swarm optimization algorithm for a vehicle routing problem with heterogeneous fleet, mixed backhauls, and time windows", *Journal of Intelligent Manufacturing*, Vol. 24, No. 4, pp. 775–789, 2013.

Benjamin, A. M. and J. Beasley, "Metaheuristics for the waste collection vehicle routing problem with time windows, driver rest period and multiple disposal facilities", *Computers & Operations Research*, Vol. 37, No. 12, pp. 2270–2280, 2010.

Berghida, M. and A. Boukra, "EBBO: an enhanced biogeography-based optimization algorithm for a vehicle routing problem with heterogeneous fleet, mixed backhauls, and time windows", *The International Journal of Advanced Manufacturing Technology*, Vol. 77, No. 9-12, pp. 1711–1725, 2015.

Bräysy, O. and M. Gendreau, "Vehicle routing problem with time windows, Part I: Route construction and local search algorithms", *Transportation Science*, Vol. 39, No. 1, pp. 104–118, 2005.

Buhrkal, K., A. Larsen and S. Ropke, "The waste collection vehicle routing problem with time windows in a city logistics context", *Procedia-Social and Behavioral Sciences*, Vol. 39, pp. 241–254, 2012.

Cheung, R. K. and D. D. Hang, "Multi-attribute label matching algorithms for vehicle routing problems with time windows and backhauls", *IIE Transactions*, Vol. 35, No. 3, pp. 191–205, 2003.

Clarke, G. and J. W. Wright, "Scheduling of vehicles from a central depot to a number of delivery points", *Operations Research*, Vol. 12, No. 4, pp. 568–581, 1964.

Coelho, L. C., J.-P. Gagliardi, J. Renaud and A. Ruiz, “Solving the vehicle routing problem with lunch break arising in the furniture delivery industry”, *Journal of the Operational Research Society*, Vol. 67, No. 5, pp. 743–751, 2016.

Gélinas, S., M. Desrochers, J. Desrosiers and M. M. Solomon, “A new branching strategy for time constrained routing problems with application to backhauling”, *Annals of Operations Research*, Vol. 61, No. 1, pp. 91–109, 1995.

Goetschalckx, M. and C. Jacobs-Blecha, “The vehicle routing problem with backhauls”, *European Journal of Operational Research*, Vol. 42, No. 1, pp. 39–51, 1989.

hostingfacts.com, *Internet Statistics & Facts (Including Mobile) for 2020*, 2020, https://hostingfacts.com/internet-facts-stats/, accessed in August 2020.

Ioannou, G., M. Kritikos and G. Prastacos, “A greedy look-ahead heuristic for the vehicle routing problem with time windows”, *Journal of the Operational Research Society*, Vol. 52, No. 5, pp. 523–537, 2001.

Kim, B.-I., S. Kim and S. Sahoo, “Waste collection vehicle routing problem with time windows”, *Computers & Operations Research*, Vol. 33, No. 12, pp. 3624–3642, 2006.

Koç, Ç. and G. Laporte, “Vehicle routing with backhauls: Review and research perspectives”, *Computers & Operations Research*, Vol. 91, pp. 79–91, 2018.

Kontoravdis, G. and J. F. Bard, “A GRASP for the vehicle routing problem with time windows”, *ORSA Journal on Computing*, Vol. 7, No. 1, pp. 10–23, 1995.

Markov, I., S. Varone and M. Bierlaire, “Integrating a heterogeneous fixed fleet and a flexible assignment of destination depots in the waste collection VRP with intermediate facilities”, *Transportation Research Part B: Methodological*, Vol. 84, pp. 256–273, 2016.

Nagy, G. and S. Salhi, "Heuristic algorithms for single and multiple depot vehicle routing problems with pickups and deliveries", *European Journal of Operational Research*, Vol. 162, No. 1, pp. 126–141, 2005.

Potvin, J.-Y., C. Duhamel and F. Guertin, "A genetic algorithm for vehicle routing with backhauling", *Applied Intelligence*, Vol. 6, No. 4, pp. 345–355, 1996.

Queiroga, E., Y. Frota, R. Sadykov, A. Subramanian, E. Uchoa and T. Vidal, "On the exact solution of vehicle routing problems with backhauls", *European Journal of Operational Research*, 2020.

Reimann, M. and H. Ulrich, "Comparing backhauling strategies in vehicle routing using ant colony optimization", *Central European Journal of Operations Research*, Vol. 14, No. 2, pp. 105–123, 2006.

Røpke, S. and D. Pisinger, "A Unified Heuristic for a Large Class of Vehicle Routing Problems with Backhauls", *Diku Tech. Report 04/14*, University of Copenhagen, Institute of Computer Science, 2004

Ropke, S. and D. Pisinger, "A unified heuristic for a large class of vehicle routing problems with backhauls", *European Journal of Operational Research*, Vol. 171, No. 3, pp. 750–775, 2006.

Sahoo, S., S. Kim, B.-I. Kim, B. Kraas and A. Popov Jr, "Routing optimization for waste management", *Interfaces*, Vol. 35, No. 1, pp. 24–36, 2005.

Solomon, M. M., "Algorithms for the vehicle routing and scheduling problems with time window constraints", *Operations Research*, Vol. 35, No. 2, pp. 254–265, 1987.

Statista, *Global retail e-commerce market size 2014-2023,* 2020, https://www.staista.com/statistics/379046/worldwide-retail-e-commerce-sales/, accessed in August 2020.

Sze, S. N., "A study on multi-trip vehicle routing problem with time windows and meal break considerations", 2011.

Taillard, É., P. Badeau, M. Gendreau, F. Guertin and J.-Y. Potvin, "A tabu search heuristic for the vehicle routing problem with soft time windows", *Transportation Science,* Vol. 31, No. 2, pp. 170–186, 1997.

Tarantilis, C. D., A. K. Anagnostopoulou and P. P. Repoussis, "Adaptive path relinking for vehicle routing and scheduling problems with product returns", *Transportation Science,* Vol. 47, No. 3, pp. 356–379, 2013.

Thangiah, S. R., J.-Y. Potvin and T. Sun, "Heuristic approaches to vehicle routing with backhauls and time windows", *Computers & Operations Research,* Vol. 23, No. 11, pp. 1043–1057, 1996.

Turkey, D., *Türkiye'de E-Ticaret 2019 Pazar Büyüklüğü*, 2020, https://www2.deloitte.com/tr/tr/pages/technology-media-and-telecommunications/articles/turkiyede-e-ticaret-2019-pazar-buyuklugu.html, accessed in August 2020.

Vidal, T., T. G. Crainic, M. Gendreau and C. Prins, "A unified solution framework for multi-attribute vehicle routing problems", *European Journal of Operational Research,* Vol. 234, No. 3, pp. 658–673, 2014.

Wu, W., Y. Tian and T. Jin, "A label based ant colony algorithm for heterogeneous vehicle routing with mixed backhaul", *Applied Soft Computing*, Vol. 47, pp. 224–234, 2016.

Zhong, Y. and M. H. Cole, "A vehicle routing problem with backhauls and time windows: a guided local search solution", *Transportation Research Part E: Logistics and Transportation Review*, Vol. 41, No. 2, pp. 131–144, 2005.

APPENDIX A: THE OPTIMAL SOLUTION PRODUCED BY MILP FOR THE INSTANCE C206 WITH 25 NODES AND 50% PICKUP CUSTOMERS: TOUR1

Tour1								
Shift						1		
Shift Start Time						0		
Shift End Time						1817		
Break Start Time						909		
Break End Time						1031		
Total Delivery Load on the Vehicle at the departure from the depot						50		
Total Pickup Load on the Vehicle at the return to the depot						100		
Break Node						1		
Node Id	Service Start Time	Earliest Service Start Time	Latest Service Start Time	Service Type (Delivery (1) or Pickup (2))	Demand	Pickup Load on the Vehicle	Delivery Load on the Vehicle	Total Load on the Vehicle
5	15.13	15.00	402.00	1	10	0	40	40
22	62.23	12.00	505.00	2	20	20	30	50
24	278.96	23.00	368.00	2	10	30	30	60
6	331.00	331.00	822.00	2	20	50	30	80
20	387.12	10.00	645.00	2	10	60	30	90
2	519.40	22.00	563.00	2	30	90	30	120
1	568.00	213.00	568.00	1	10	90	30	120
7	1090.27	965.00	1340.00	1	20	90	10	100
3	1122.39	1030.00	1463.00	1	10	90	0	90
4	1154.00	1154.00	1527.00	2	10	100	0	100

APPENDIX B: THE OPTIMAL SOLUTION PRODUCED BY MILP FOR THE INSTANCE C206 WITH 25 NODES AND 50% PICKUP CUSTOMERS: TOUR2

Tour2								
Shift						2		
Shift Start Time						1573		
Shift End Time						3390		
Break Start Time						2482		
Break End Time						2604		
Total Delivery Load on the Vehicle at the departure from the depot						190		
Total Pickup Load on the Vehicle at the return to the depot						120		
Break Node						12		
Node Id	Service Start Time	Earliest Service Start Time	Latest Service Start Time	Service Type (Delivery (1) or Pickup (2))	Demand	Pickup Load on the Vehicle	Delivery Load on the Vehicle	Total Load on the Vehicle
23	1603.53	1519.00	1926.00	1	10	0	180	180
18	1640.59	1560.00	2083.00	2	20	20	180	200
19	2109.17	1689.00	2144.00	1	10	20	170	190
17	2143.00	2143.00	2638.00	1	20	20	150	170
15	2176.39	2077.00	2514.00	1	40	20	110	130
16	2264.00	1763.00	2264.00	2	40	60	110	170
14	2306.00	1846.00	2365.00	2	10	70	110	180
12	2412.00	1985.00	2412.00	2	20	90	110	200
13	2659.00	2310.00	2659.00	1	30	90	80	170
25	2787.00	2380.00	2787.00	1	40	90	40	130
9	2822.21	2385.00	2976.00	1	10	90	30	120
11	2952.00	2603.00	2952.00	1	10	90	20	110
10	2983.00	2628.00	3113.00	2	10	100	20	120
8	3029.08	2653.00	3280.00	2	20	120	20	140
21	3074.74	2675.00	3288.00	1	20	120	0	120